Structure-Preserving Variational Integration for Constrained Optimal Control of a Wheeled Inverted Pendulum

Azeez Mohammed, Ciccarelli Emanuele, Viscione Michele

October 7, 2022

1 Introduction

The Wheeled Inverted Pendulum (WIP) is a nonholonomic, underactuated mechanical system, and has been popularized commercially as the Segway. The focal point of this work is to derive a discrete-time model of the WIP system using discrete mechanics and generating optimal trajectories by solving a discrete-time constrained optimal control problem. Nextly, we employ a closed-loop linear–quadratic regulator (LQR) in a dual control architecture to track the generated optimal trajectory which is provided as a reference to the robot. The LQR in feedback mode is designed to mitigate noise and disturbances for ensuring stable motion of the WIP system.

Synthesizing discrete-time control signals for mechanical systems with state and control constraints, while preserving the underlying mechanical structure in the process of discretization, is a challenging problem. Quite often, the continuous time controller is either discretized after design or, sometimes, the system model is discretized to start with, by means of a standard discretization scheme, with the control design proceeding thereafter. These procedures are approximations, and in the latter, **one loses the mechanical nature of the system when discretization is performed**. A faithful discretization scheme followed by a computationally tractable control law synthesis, which respects multiple (state and control) constraints, is, therefore, most desirable.

The Variational Integrator, a geometry-preserving integration scheme obtained from discrete mechanics, is a reasonable and efficient solution to this problem. They are numerical integrators for Hamiltonian systems which also allow the preservation of momentum as well as its symplectic structure, while exhibiting an energy conservation property over long time intervals as demonstrated in our simulation results. In particular in section 3, thanks to a toy example, we will expose how the Variational Integration method allows us to better conserve the structure of the configuration space as well as the energy in comparison to other discretization methods (e.g. Runge Kutta). The core idea is to preserve the Lagrangian's symmetries under discretization and therefore maintaining the conservation rules thanks to Noether's theorem.

In section 4 we will then propose the use of the Variational Integrator scheme for the dynamics of the wheeled inverted pendulum as in [APA+19] and [PBC17], where an implicit discrete model representation is defined. Moreover, we will push further the line by finding an explicit closed form formulation for this model, which has of course many benefits. In particular, this will allow us to lower the computational effort of IPOPT, the nonlinear optimization tool that we used to plan our optimal trajectories, as well as granting us the possibility to verify the influence of the state constraints errors within our simulations. Finally, in the last section 6 we will empower our control scheme with the aforementioned LQR controller with the aim of lowering the the sim-to-real gap by robustly reject unaccounted disturbances.

All the code used for the simulations within the following sections can be found in this GitHub repository.

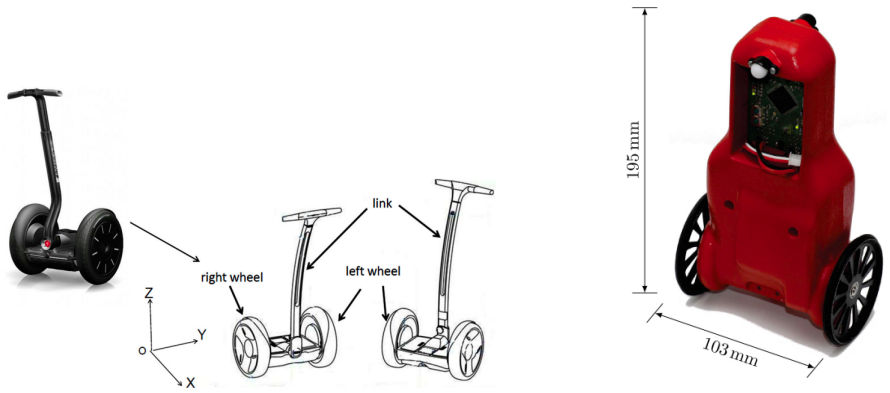

Figure 1: To the left is the commercial version of the WIP known as Segway

2 WIP Modeling

The WIP, consists of a vertical body with two coaxial driven wheels. The system belongs to the class of underactuated mechanical systems, since the number of control inputs is less than the number of degrees of freedom, in addition, the system has nonholonomic constraints that arise due to the pure rolling assumption on the wheels.

The WIP finds many applications, that include baggage transportation, commuting and navigation. It has gained interest in the past several years due to its maneuverability and simple construction. Other robotic systems based on the WIP are fastly becoming popular as well in the robotics community for human assistance and transportation for which there are commercially available models of the Segway robot used for human transportation.

2.1 Modelling

The WIP, consists of a body of mass m_b mounted on wheels of radius r_w and at a height l from the wheel rotation axis. A pair of wheels of mass m_w each, are mounted on the base of the body with a distance $2d_w$ between them, with the wheels being able to rotate independently. The actuation of both wheels in the same direction generates a forward (or backward) motion; opposite wheel velocities lead to a turning motion around the vertical axis. For these type of systems, one of the control objectives is to stabilize the body in the upward position via back and forth motion of the system since the upward position of the body represents an unstable equilibrium which needs to be stabilized.

The configuration variables of the system are:

- $(x, y) \in R^2$: the coordinates of the origin of the body-fixed frame in the horizontal plane of the inertial frame
- $\theta \in S^1$: the heading angle which is the angle of the wheel rotation axis with the x-axis or the y-axis in the inertial frame
- $\alpha \in S^1$: the tilt angle of the body which is the angle of the body z-axis with the horizontal plane in the inertial frame
- ϕ_R and $\phi_L \in S^1$: the relative rotation of individual wheels w.r.t. the body-fixed frame about the wheel rotation axis.

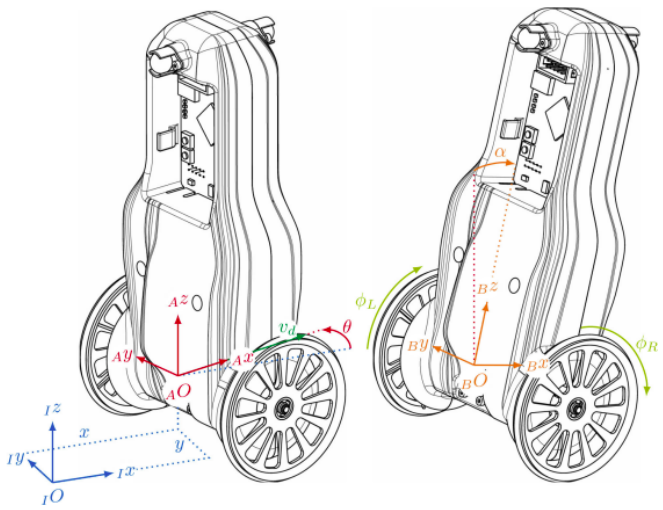

Figure 2: Coordinate systems and parameters of the WIP

Having defined these quantities, the configuration space Q of the system is $SE(2) \times S^1 \times S^1 \times S^1$, with a state $q := (x, y, \theta, \alpha, \phi_L, \phi_R) \in Q$.
As aforementioned, the system motion is restricted by nonholonomic (nonintegrable) constraints that arise due to the no-slip conditions on the wheels, which in this case are the no lateral sliding and pure rotation without slipping constraints. These constraints do not restrict the configuration space Q in which the dynamics evolve, but the motion direction at a given point:
The pure rolling motion of the wheels is given by

$$\begin{cases} \dot{x_R}\cos\theta(t) + \dot{y_R}\cos\theta(t) = r_w\dot{\phi_R} \\ \dot{x_L}\cos\theta(t) + \dot{y_R}\cos\theta(t) = r_w\dot{\phi_R} \end{cases} \tag{1}$$

and the no side-slip constraints are given by

$$\begin{cases} -\dot{x_R}\sin\theta(t) + \dot{y_R}\cos\theta(t) = 0 \\ -\dot{x_L}\sin\theta(t) + \dot{y_R}\cos\theta(t) = 0 \end{cases} \tag{2}$$

Where $(x_L, y_L) \in R^2$ and $(x_R, y_R) \in R^2$ are the the left and right wheel's position on the x-y plane in the spatial coordinates. Furthermore they can be expressed as:

- $x_L(t) := x(t) - d_w \sin\theta(t)$
- $y_L(t) := y(t) + d_w \cos\theta(t)$
- $x_R(t) := x(t) + d_w \sin\theta(t)$
- $y_R(t) := y(t) - d_w \cos\theta(t)$

Taking into consideration the aforementioned expressions, the four constraints hence can be compressed to the form :

$$\begin{cases} \dot{x(t)} - \dfrac{r_w}{2}\cos\theta(t)(\dot{\phi_R} + \dot{\phi_L}) = 0 \\ \dot{y(t)} - \dfrac{r_w}{2}\sin\theta(t)(\dot{\phi_R} + \dot{\phi_L}) = 0 \\ \dot{\theta(t)} - \dfrac{r_w}{2d_w}\theta(t)(\dot{\phi_R} - \dot{\phi_L}) = 0 \end{cases} \tag{3}$$

Having defined the constraints which the system is subject to, we can now derive the Lagrangian of the WIP system.
In order to charcterize the Lagrangian of the system with $L = T - V$, to do so we need to define the

kinetic energy T and potential energy V.
The total kinetic energy of the system T is the sum of kinetic energy of the main body and the wheels. Furthermore, $T = T_b + T_w$ where T_b represents the kinetic energy of the main body and is given by :

$$T_b(q, \dot{q}) = \frac{1}{2}(m_b v_b^T v_b) + (\omega_b^T I_B \omega_b) \tag{4}$$

m_b represents the mass of the main body, v_b is the translational velocity, I_B is the inertia of the main body with respect to its center of mass in the body-fixed frame and ω_b the angular velocity of the body.

In particular $I_B := diag(I_{Bxx}, I_{Byy}, I_{Bzz})$, $v_b := \begin{pmatrix} \dot{x} + l\dot{\alpha}\cos\alpha\cos\theta - l\dot{\alpha}\sin\alpha\sin\theta \\ \dot{y} + l\dot{\alpha}\cos\alpha\cos\theta + l\dot{\alpha}\sin\alpha\sin\theta \\ l\dot{\alpha}\sin\alpha \end{pmatrix}$

and $\omega_b := \begin{pmatrix} \dot{\theta}\sin\alpha \\ \dot{\alpha} \\ \dot{\theta}\cos\alpha \end{pmatrix}$

Now let m_w be the mass of the wheels, also let $v_{w,L}$, $v_{w,R}$ be the translational velocity of the right and the left wheel's center of mass, $\omega_{w,R}$ and $\omega_{w,L}$ be the angular velocity of the left and right wheels and $I_W := diag(I_{Wxx}, I_{Wyy}, I_{Wzz})$ denote the inertia of the wheels with respect to its center of mass in the body-fixed frame.
Then T_w represents the kinetic energy of the wheels and is given by :

$$T_w(q, \dot{q}) = \frac{1}{2}(m_w v_{w,R}^T v_{w,R} + \omega_{w,R}^T I_W \omega_{w,R}) + (m_w v_{w,L}^T v_{w,L} + \omega_{w,L}^T I_W \omega_{w,L}) \tag{5}$$

where $v_{w,L} = \begin{pmatrix} \dot{x} - d_w\dot{\theta}\cos\theta \\ \dot{y} - d_w\dot{\theta}\sin\theta \\ 0 \end{pmatrix}$ and $v_{w,R} = \begin{pmatrix} \dot{x} + d_w\dot{\theta}\cos\theta \\ \dot{y} + d_w\dot{\theta}\sin\theta \\ 0 \end{pmatrix}$

The potential energy of the system V is gravitational and is expressed as :

$$V(q, \dot{q}) = m_b g l \cos\alpha \tag{6}$$

In a mechanical system with nonholonomic constraints, the n-dimensional manifold Q is the configuration space, its tangent bundle TQ is the velocity phase space and a smooth (nonintegrable) distribution $D \in TQ$ represents the constraints.
The Lagrangian L can then be visualized as a map from the tangent bundle to the space of real numbers

$$TQ \ni (q, \dot{q}) \to L(q, \dot{q}) = T(q, \dot{q}) - V(q, \dot{q}) \tag{7}$$

3 Variational Integrator

Discretizing continuous dynamics or differentially defined systems are well known problems and the search for their solutions is mandatory whenever we want to simulate or control systems via digital hardware. A common baseline for the numeric integration algorithms is the 4-th order Runge-Kutta method (RK4) while the naivest approach is probably the forward Euler Method (FEM). Generally, through this discretization procedure, we lose the symmetries of the original Lagrangian and therefore, by Noether's theorem, the conservation rules that hold for the original continuous time system. The purpose of the Variational Integrator (VI) method is to fill this gap by defining a new discrete Hamilton's Principle instead of discretizing the dynamics that satisfies the original continuous one. We quote [KTG16] for a nice summarized rephrasing of this concept:

"It is not the dynamical equations that are discretised but the building blocks of the underlying field theory"

The resulting equations obtained from the VI will preserve momenta related to this discrete Lagrangian's symmetries as well as the related discrete symplectic form (generalization of the phase space) [LMOW04]. In particular, the Lie Group Variational Integrators are numerical methods that retain the geometry-preserving properties of the VI while allowing a computationally efficient evolution on Lie Groups [Leo07]. Therefore, once we have defined parts of our dynamics as an exponential

update map from an element of the Lie Algebra to its related Lie Group (see [IMKNZ00], Def 2.1 and Theorem 2.8) we can build the Discrete Action Principle on this representation and, by construction, force our dynamics to remain in that group while preserving the related general momenta. We will not probe too much in the computational details of this procedure, we refer the interested readers to the aforementioned references for further details.

Toy example: Pendulum What we will do instead, partially inspired by the work of [SK22], is to present a brief toy example to showcase the behavior of this method with respect to the previously mentioned FEM and RK4 numeric algorithms. In particular, we have integrated the dynamics of a pendulum in $\mathbb{R}^2$ coordinates, solutions of

$$mL^2\ddot{q} + mL^2||\dot{q}||^2 q + mgL(I_{2x2} - qq^T)e_2 = 0, \quad q \in \mathbb{R}^2 \tag{8}$$

The model obtained by the FEM is defined as

$$q_{k+1} = 2q_k - q_{k-1} - (q_k^T q_k - 2q_k^T q_{k-1} + q_{k-1}^T q_{k-1})q_{k-1} - \frac{gh^2}{L}(I_{2x2} - q_{k-1}q_{k-1}^T)e_2 \tag{9}$$

while the related Variational Integrated model is the following

$$q_{k+1} = [q_k^T S q_{k-1} - \frac{h^2 g}{L} e_2^T S q_k \;\; 1][Sq_k \;\; q_k]^{-1} \tag{10}$$

where S is the skew-symmetric matrix used for the definition of the exponential map and $e_2 = (0,1)^T$ (see [SK22] for further details).

Now observe fig.3, there we can see how the FEM is not capable of handling 10 seconds of pendulum simulation with discrete time step $h = 0.001$ without losing both the phase portrait structure and the energy conservation. To make this method work we need to significantly lower the time step length.

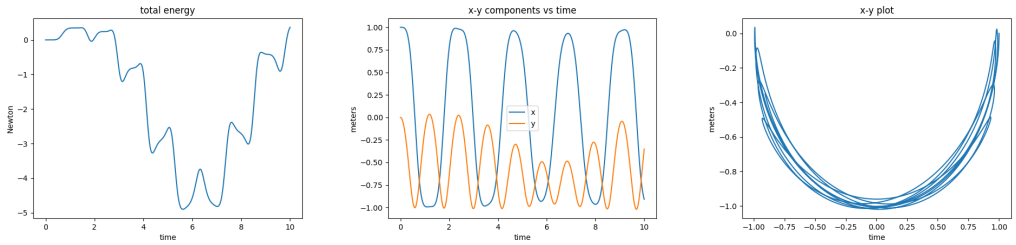

Figure 3: First graph represents the total energy of the system, the second one shows the x and y coordinates against t while the last one represents the phase portrait $x - y$. Simulation results of the model (9) using forward Euler. Ten seconds are simulated with a discrete time step $h = 0.001$ s. The structure begins to deform while the energy is oscillating even with this small step size h. Few more seconds and both the energy and the structure will collapse

For this precise reason, the real comparison will be among the RK4 algorithm (fig.4) and the VI (fig.5). The two figures show the RK4 pendulum simulation of 100 seconds with $h = 0.01$ s and the VI method with 2000 seconds simulated under the same h. Both algorithms preserve the phase portrait structure for small enough h, although the VI is far more resilient, but the notable difference lies in how well the VI conserves the total energy (that must be 0 in this case) with respect to the RK4 method. Even way longer simulations can be handled gracefully by the VI without losing the energy conservation, moreover observe that the fluctuations are so small that we are in the regime of Python's numerical approximations. To further emphasize this difference, we report in fig.6 (RK4) and fig.7 (VI) the energy plots of a 10 seconds simulation for the pendulum as h varies.

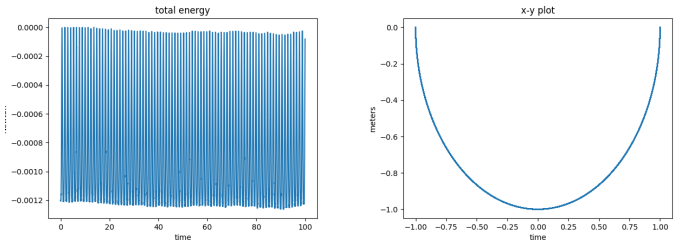

Figure 4: Simulation results of the model (9) using RK4. First graph represents the total energy of the system, the second one shows the phase portrait $x - y$. One-hundred seconds are simulated with a discrete time step $h = 0.01$ s.

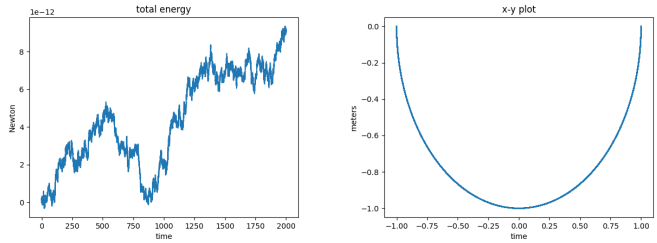

Figure 5: Simulation results of the model (10), which uses the Variational Integrator. First graph represents the total energy of the system, the second one shows the phase portrait $x-y$. Two-thousand seconds are simulated with a discrete time step $h = 0.01$ s. Even with this very long simulation time and with a reasonable step size we can clearly see how the structure is completely preserved and the energy still remain negligible.

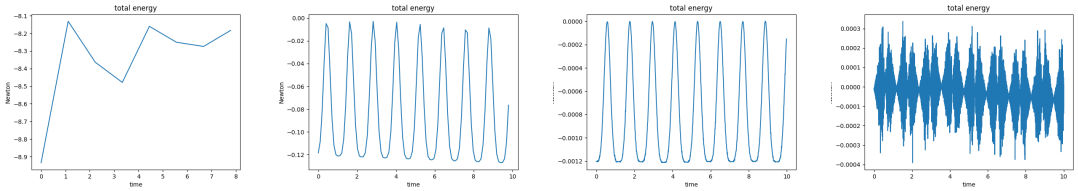

Figure 6: Energy profiles of the RK4 numeric integration, 10 seconds are simulated with a discrete step length h of 1 s, 0.1 s, 0.01, 0.001 s respectively. The bigger the step, the worse the energy is conserved

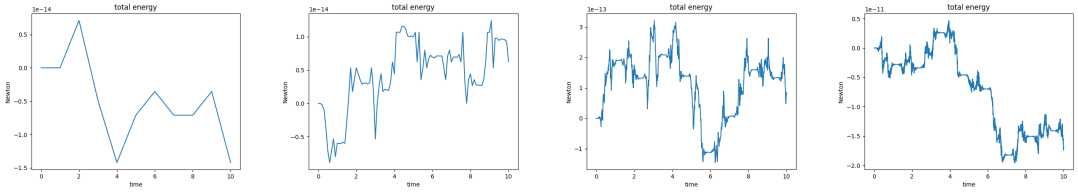

Figure 7: Energy profiles of the Variational Integrated model (10), 10 seconds are simulated with a discrete step length h of 1 s, 0.1 s, 0.01, 0.001 s respectively.

4 Structure-Preserving Integration of the WIP model

Now that we have a general understanding on what is a Variational Integrator we will apply this methodology to the case of the WIP model. It would be out of the scope of this work to probe too

deeply into the theoretical features behind the construction of this model, we then limit ourselves to summarize the steps from [APA+19] and [PBC17] in order to obtain a closed form implicit model for the WIP under the VI technique. Be $q \in Q = G \times M$, $q = ((x, y, \theta), (\alpha, \phi_1, \phi_2), (v_\alpha, v_{1_1}, v_{\phi_2}))^T = (g, s, v)^T$ the elements of the configuration space, where G is a Lie group and $Q = G \times M$ a principal fiber bundle. The first step is to define the reduced Lagrangian

$$\begin{aligned} L^\flat(s, v, \xi) := \frac{1}{2}(m_b + 2m_w)(\xi_1^2 + \xi_2^2)\frac{1}{2}I_\theta(\alpha)\xi_3^2 + \frac{1}{2}(m_b b^2 + I_{B_{yy}})v_\alpha^2 + \\ + \frac{1}{2}I_{W_{yy}}(v_{\phi_1}^2 + v_{\phi_2}^2) + m_b b \sin\alpha \xi_1 \xi_2 + m_b b \cos\alpha \xi_1 v_\alpha + m_b b g \cos\alpha \end{aligned} \tag{11}$$

where $\xi = (v_x \cos\theta + v_y \sin\theta, -v_x \sin\theta + v_y \cos\theta, v_\theta)$. Intuitively, this Lagrangian acts on a reduced space that naturally embeds the nonholonomic constraint that characterizes our model. More about Lagrangian reduction for nonholonomic systems can be found in [CMR01].

To compute the coordinates for this reduced space it is useful to introduce a principal connection on the bundle $Q \to Q/G$, where a reasonable choice is to exploit a mechanical connection ([KM97], section 3.1). In our case we will introduce the nonholonomic connection, whose local form is defined by $\xi + \mathbb{A}(s)v = 0$, which establishes the effect of the nonholonomic constraints within our reduced space. The matrix $\mathbb{A}(s)$ is defined as

$$\mathbb{A}(s) = \begin{pmatrix} 0 & -r_w & -r_w \\ 0 & 0 & 0 \\ 0 & \frac{r_w}{d_w} & -\frac{r_w}{d_w} \end{pmatrix} \tag{12}$$

This has a crucial role for the definition of the exponential map that drives the dynamics of g, in the sense that the matrix (12) maps the velocities v_s to corresponding elements of the Lie algebra that defines velocities over the fiber space G (see [APA+19], Appendix A.B). In this way, we establish a *connection* between elements in the tangent space of M and velocities of elements in the Lie group G.

4.1 Implicit Model

Now that we have seen two of the most important ingredients, we are ready to define the implicit model for the WIP under Variational Integration in the same manner as [APA+19].

$$g_{k+1} = g_k e(-h\mathbb{A}(s_k)v_k)$$

$$s_{k+1} = s_k + hv_k$$

$$\frac{\partial L_k^\flat}{\partial v} - h\frac{\partial L_k^\flat}{\partial s} - (De^{-1}(-v_k) \circ \mathbb{A}(s_k))^*\left(\frac{\partial L_k^\flat}{\partial \xi}\right) = \frac{\partial L_{k-1}^\flat}{\partial v} - (De^{-1}(-v_{k-1}) \circ \mathbb{A}(s_{k-1}))^*\left(\frac{\partial L_{k-1}^\flat}{\partial \xi}\right) + h\mathcal{F}_{k-1}$$

where $e(\cdot)$ is the exponential map,

$$De^{-1}(v_k) := Te^{-1}(e(-h\mathbb{A}(s_k)v_k)) \circ e(-h\mathbb{A}(s_k)v_k)$$

and

$$L_k^\flat := L^\flat(s_k, (s_{k+1} - s_k)/h, e^{-1}(g_k^{-1}g_{k+1})/h)$$

Note that to find the complete state - in particular, $v(k+1)$ - it is necessary to solve the third implicit equation, a discretized version of the reduced Euler-Lagrangian equations.

A more refined representation of this set of equations can be found in [PBC17]. The next section will use this formulation as a reference for the implicit model of the WIP.

$$g_{k+1} = g_k e^{-h\mathbb{A}v_k} \tag{13}$$

$$s_{k+1} = s_k + hv_k \tag{14}$$

$$\mathbb{M}(\alpha_{k+1})v_{k+1} - h\mathbb{C}(\alpha_{k+1}, v_{k+1}) = \mathbb{M}(\alpha_k)v_k + h\tau_k \tag{15}$$

with

$$\mathbb{M}(\alpha) = \begin{pmatrix} (m_b b^2 + I_{Byy}) & (m_b b r_w \cos\alpha) & (m_b b r_w \cos\alpha) \\ k_{12}(m_b b r_w \cos\alpha) & \mathbb{H}(\alpha) + I_{Wyy} & \mathbb{K}(\alpha) \\ (m_b b r_w \cos\alpha) & \mathbb{K}(\alpha) & \mathbb{H}(\alpha) + I_{Wyy} \end{pmatrix}$$

$$\mathbb{K}(\alpha) := {r_w}^2((m_b + 2m_w) - \frac{I_\theta(\alpha)}{2{d_w}^2}$$

$$\mathbb{H}(\alpha) := {r_w}^2((m_b + 2m_w) + \frac{I_\theta(\alpha)}{2{d_w}^2}$$

$$\mathbb{C}(\alpha, v) := (\frac{{r_w}^2}{2{d_w}^2}(I_{Bxx} - I_{Bzz} + m_b b^2)\sin 2\alpha(v_{\phi 1} - v_{\phi 2})^2 - m_b b r_w \sin\alpha v_\alpha(v_{\phi 1} + v_{\phi 1}) + m_b b g \sin\alpha, 0, 0)^T$$

$$I_\theta(\alpha) := 2I_{Wzz} + I_{Bzz}\cos^2\alpha + 2m_w d^2 + (I_{Bxx} + m_b b^2)\sin^2\alpha$$

4.2 Explicit Model

One of the limitations of possessing only a non-trivial implicit representation for this model is that, in practice, you can determine its dynamics only with the help of some numerical optimization software. To overcome this lack of versatility and to obtain something that we can simulate efficiently we successfully found a closed form explicit representation for this model. Looking at the implicit form previously defined, the update maps for g_k and s_k are itself explicit so the real explicitation regards only the $v = (v_\alpha, v_{\phi_1}, v_{\phi_2})^T$ component, obtainable from the matricial equation (15). The update maps for these three components are

$$v_{\phi_1}(k+1) = \frac{\Psi(k)v_\alpha(k) - \Psi(k+1)v_\alpha(k+1)}{\delta} + \frac{v_{\phi_1}(k) + v_{\phi_2}(k)}{2} + \frac{h}{2\delta}(\tau_1 + \tau_2) + \frac{D}{2} \tag{16}$$

$$v_{\phi_2}(k+1) = \frac{\Psi(k)v_\alpha(k) - \Psi(k+1)v_\alpha(k+1)}{\delta} + \frac{v_{\phi_1}(k) + v_{\phi_2}(k)}{2} + \frac{h}{2\delta}(\tau_1 + \tau_2) - \frac{D}{2} \tag{17}$$

$$v_\alpha(k+1) = \frac{2\mu}{\rho + \sqrt{\rho^2 + 4\lambda\mu}} \tag{18}$$

where

$$\Psi(k) = m_b b r_w \cos\alpha_k$$
$$\Phi(k) = m_b b r_w \sin\alpha_k$$
$$\Omega = m_b b^2 + I_{B_{yy}}$$

$$\lambda = \frac{2h}{\delta}\Psi(k+1)\Phi(k+1)$$

$$\rho = \Omega - \frac{2}{\delta}\Psi(k+1)^2 + h\Phi(k+1)\eta$$

$$\mu = Z + hC_0 - \Psi(k+1)\eta$$

The remaining unknown terms are

- D function of $\alpha(k+1)$, $\alpha(k)$, $v_{\phi_1}(k)$, $v_{\phi_2}(k)$, τ_1, τ_2
- C_0 function of D, $\alpha(k+1)$, $\Phi(k+1)$
- Z function of Ω, $\Psi(k)$, $v_{\phi_1}(k)$, $v_{\phi_2}(k)$, $v_\alpha(k)$
- η function of $\Psi(k)$, $v_\alpha(k)$, $v_{\phi_1}(k)$, $v_{\phi_2}(k)$, τ_1, τ_2

these are defined in the Appendix along with the procedure and computations to obtain these results. In fig. 8 it is shown the computational graph that gives us the dependencies to compute the next state at each time step k.

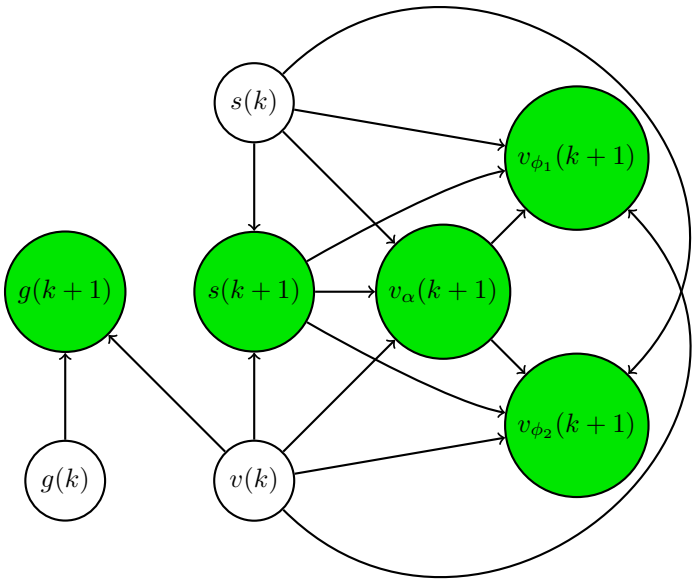

Figure 8: Computational Graph of the explicit closed form model. The green states form the complete updated state $q(k+1) = (g(k+1), s(k+1), v(k+1))^T$. The computational order that must be used to generate the next state is $g, s \to v_\alpha \to v_{\phi_1}, v_{\phi_2}$.

5 Trajectory Planning and Optimal Control

Now that we have both implicit and explicit representations for our system, we will use numeric optimization software to plan optimal trajectories that pass through prespecified waypoints at specific times while respecting some constraints on both the state and the control action. To do so, we used IPOPT, a software library for large scale nonlinear optimization, within Python. We can formulate the optimization problem as

$$\min_{(\tau_k)_{k=0}^{N-1}} C(\tau) := \sum_{k=0}^{N-1} \frac{h}{2} \langle \tau_k, \tau_k \rangle \tag{19}$$

subject to

$$
\begin{cases}
\begin{cases}
g_{k+1} = g_k e^{-h\mathbb{A}v_k} \\
s_{k+1} = s_k + hv_k \\
z_{k+1} = \mathbb{M}(\alpha_{k+1})v_{k+1} - h\mathbb{C}(\alpha_{k+1}, v_{k+1}) \\
z_{k+1} = \mathbb{M}(\alpha_k)v_k + h\tau_k
\end{cases} \\
||\tau_k||_\infty \leq \mu \\
\frac{1}{2}((v_k^j)^2 - \nu^2) \leq 0 \\
\frac{1}{2}((\alpha_k)^2 - a^2) \leq 0 \\
(g_0, s_0, v_0) = (g^i, s^i, v^i) \\
(g_N, \alpha_N, v_N) = (g^f, \alpha^f, v^f)
\end{cases}
\tag{20}
$$

for $k = 1, ..., N-1$ and $j = 1, 2, 3$.

Of course, in the case of explicit model the third and forth equation of (20) are substituted with a direct constraint over v_{k+1}, analogously to g_{k+1} and s_{k+1}. A key factor for achieving a satisfactory computational efficiency was the vectorialization of the code parts related to the computation of the state constraints for both the explicit and implicit case (i.e. computing in batch the constraints for all the simulation steps). By vectorializing the code we have seen a truly relevant improvement in terms of IPOPT optimization speed, about 20-60 times depending on the problem specifications. As the intuition suggests, the vectorialized explicit model constraint is the most efficient and precise setup for the optimization problem. Another factor which we experimented with is the initialization of the optimizer and the initial states sequence. For instance, by initializing the states with trajectories which interpolate the waypoints we can greatly help the optimizer whenever the interpolating path is close to the optimal trajectory. In fig. 9 are shown the results of such an optimization task, where the WIP, starting at $\text{node}_0 = (x, y, \theta, t) = (1, 1, 0°, 0)$ has to pass through 4 waypoints: $\text{node}_1 = (4, 3, 0°, 2)$, $\text{node}_2 = (5, 1, 0°, 4)$, $\text{node}_3 = (3, -2, -30°, 6)$ and $\text{node}_4 = (1, 1, -60°, 8)$. This trajectory is performed in 8 seconds, with discrete time step $h = 0.05$ s.

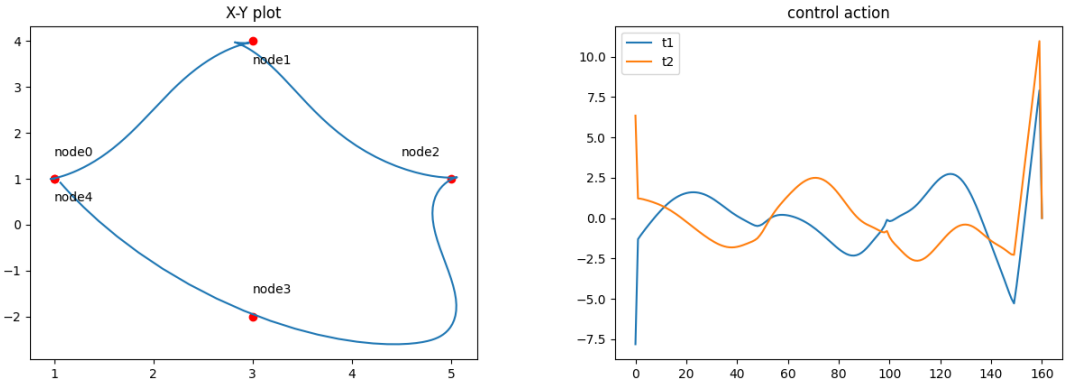

Figure 9: The right side image shows the the optimal trajectory computed by IPOPT on the X-Y plane for a 4 waypoints task (the last waypoint is the initial position) with specified heading angle values. The left side image is the corresponding control action on both the wheels.

We also used the Vpython library to develop a 3D visualizer for our planned trajectories (fig. 10).

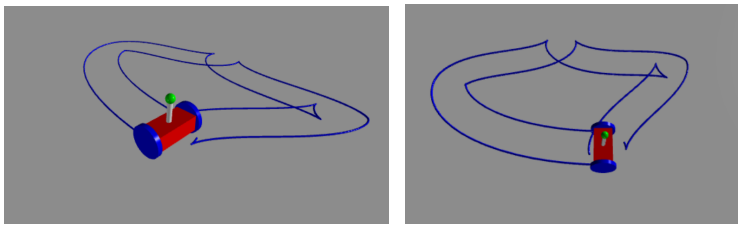

Figure 10: 3D visualization of the WIP following an optimal trajectory found by IPOPT.

Since the software finds both the state trajectory and the optimal control at the same time it is common that small errors of the state constraints sum up to the point that, especially for longer simulations, the so called optimal trajectory is not the one we would have from applying directly the just found optimal control action to the real system. We can then exploit our explicit model to verify how much the real trajectory of the model under this optimal control input differs from the optimized one found with IPOPT. In fig. 11 it shown an example of what can happen if we allow too high state constraints errors.

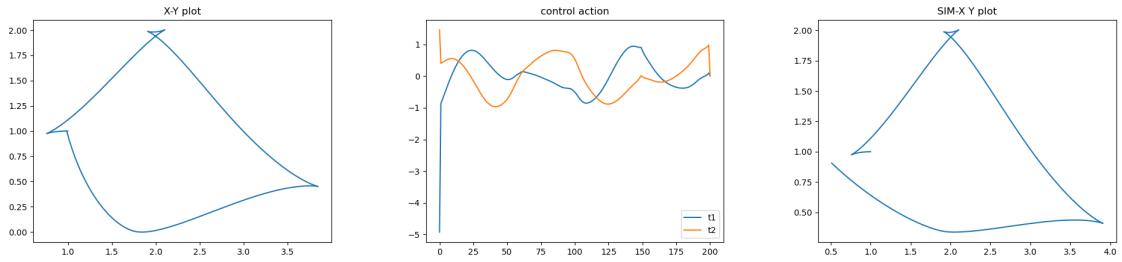

Figure 11: The right image shows the the optimal trajectory computed by IPOPT on the X-Y plane for a 4 waypoints task (the last waypoint is the initial position) with specified heading angle values. The state constraints error at the end of the optimization procedure was in the order of 10^{-2}-10^{-3}. The central image is the corresponding control action on both the wheels. The left side image is generated thanks to the explicit model and shows the real behavior of the WIP under such control action. The real trajectory do not ends onto the right waypoint!

The next step to effectively track this trajectory is to consider also the sim-to-real gap. To do so, after the constrained discrete-time optimal trajectory for the WIP is computed offline (planning-phase), we will then add an extra input $\hat{u}$ computed from an LQR controller whose objective is to mitigate unaccounted disturbances and to maintain stability of the system, as shown in figure 12. This particular controller architecture has the advantage that it is easy to implement and the overall online computation time is low since the only on-the-fly computation needed is the one accounting for the LQR controller.

6 LQR and final Control Architecture

In order to develop an LQR controller we need a linearized model of the WIP. In what follows we have used a slightly different approach than the one used in [APA+19], however as we will see, we will arrive at the same result.

6.1 Linearization of the model

First of all, using a Lagrangian approach, we can derive the nonlinear equations of motion for the WIP as shown in [Ngu20]:

$$m_{11}\ddot{x} - m_b(\dot{\theta}^2 + \dot{\alpha}^2)\sin\alpha + m_{12}\ddot{\alpha} + \frac{2}{d_v}(\frac{\dot{x}}{r_w} - \dot{\alpha}) = \frac{\tau_L + \tau_R}{r_w}$$

$$m_{21}\ddot{x} - m_{22}\ddot{\alpha} + m_{23}\dot{\theta}^2\sin 2\alpha - m_B l g \sin\alpha - 2d_v(\frac{\dot{x}}{r_w} - \dot{\alpha}) = -(\tau_L + \tau_R)$$

$$m_{33}\ddot{\theta} + (m_b l\dot{x} - m_{32})\dot{\theta}\sin\alpha + \frac{d_v\dot{\theta}(2d_w)^2}{2r_w^2} = \frac{(\tau_R - \tau_L)2d_w}{2r_w}$$

Where

$$m_{11} = m_b + 2m_w + \frac{2I_{Wyy}}{r_w^2}$$

$$m_{12} = m_{21} = m_b l \cos\alpha$$

,

$$m_{22} = (m_b l^2 + I_{Byy})$$

$$m_{23} = \frac{(I_{Bzz} - I_{Bxx} - m_b l^2)}{2}$$

$$m_{32} = 2(I_{Bzz} - I_{Bxx} - m_b l^2)$$

$$m_{33} = I_{Bzz} + 2I_{Wzz} + 2\frac{m_w(2d_w)^2}{2} + \frac{I_{Wyy}(2d_w)^2}{2r_w^2} - (I_{Bzz} - I_{Bxx} - m_b l^2)\alpha$$

Now, let define state variables and inputs as

$$\mathbf{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]^T = [x\ \theta\ \psi\ \dot{x}\ \dot{\theta}\ \dot{\psi}]^T$$

and

$$\mathbf{u} = [u_1\ u_2]^T = [\tau_1\ \tau_2]^T$$

Note that, without loss of generality, we can take the state $\mathbf{x}$ without the variable y, indeed, as we can see in [APA+19], after the linearization at the heading angle $\theta = 0$ the state y is decoupled from the other states and control inputs. So, if we linearize at the equilibrium point $\mathbf{x} = \mathbf{0}$, we get the following linear system

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & a_{42} & 0 & a_{44} & a_{45} & 0 \\ 0 & a_{52} & 0 & a_{54} & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ b_{41} & b_{42} \\ b_{51} & b_{52} \\ b_{61} & b_{62} \end{bmatrix} \mathbf{u} \tag{21}$$

Where

$$\Lambda = 2I_{Byy}I_{Wyy} + 2I_{Wyy}m_b l^2 + I_{Byy}m_b r_w^2 + 2I_{Byy}m_w r_w^2 + 2m_b l^2 m_w r_w^2$$

$$a_{42} = -\frac{(m_b g l)^2 g}{\Lambda}$$

$$a_{44} = -\frac{2d_v[(m_b l^2 + I_{Byy}) + m_b l r_w]]}{\Lambda}$$

$$a_{45} = \frac{2d_v r_w(m_b l^2 + I_{Byy}) + 2d_v m_b l^2}{\Lambda}$$

$$a_{52} = \frac{m_b g l(2I_{Wyy} + m_b r_w^2 + 2m_w r_w^2)}{\Lambda}$$

$$a_{54} = \frac{2d_v m_b l + 2d_v(2I_{Wyy} + m_b r_w^2 + 2m_w r_w^2)/r_w}{\Lambda}$$

$$a_{55} = \frac{2d_v(2I_{Wyy} + m_b r_w^2 + 2m_w r_w^3) + 2d_v m_b l^2)}{\Lambda}$$

$$a_{66} = -\frac{d_v(2d_w)^2)}{2I_{Wyy}(2d_w)^2) + (2I_{Bzz} + 4I_{Wzz} + m_w(2d_w)^2)r_w^2}$$

$$b_{41} = b_{42} = \frac{r_w[(m_b l^2 + I_{Byy}) + m_b l r_w]}{\Lambda}$$

$$b_{51} = b_{52} = -\frac{2I_{Wyy}m_b r_w^2 + 2m_w r_w^2 + m_b l r_w}{\Lambda}$$

$$b_{61} = -b_{62} = -\frac{2d_w r_w}{I_{Wyy}(2d_W)^2 + (2I_{Bzz} + 4I_{Wzz} + (2d_w)^2 m_w)r_w}$$

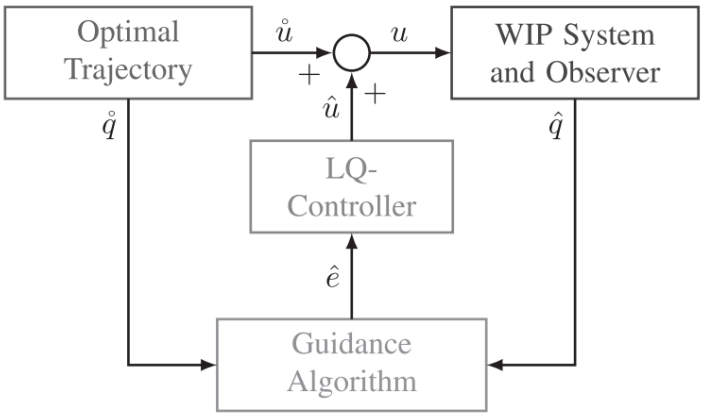

Figure 12: Closed-loop WIP system

6.2 Computation of the optimal gain

From the linearized model (21), a LQR controller is designed as done in [APA+19] by setting the weighting matrix

$$Q_Y := diag(\begin{bmatrix}1500 & 350 & 0.1 & 0 & 0 & 1\end{bmatrix})$$

for the states and

$$R_Y := I_{2x2}$$

The optimal feedback is then $u = -K\mathbf{x}$, where K is

$$K = R_Y^{-1} B^T P$$

Where P is the solution of the classic Riccati equation.

6.3 General control scheme

The linearized system does not have access to the states (x, y), but only the traveled forward distance d is available to the controller. Therefore, in order to facilitate motion of the system on the $x - y$ plane, we need a guidance algorithm that will generate the necessary control actions.
In particular, as in [APA+19], the guidance algorithm calculates the control errors for the controller in distance e_d and angle e_θ based on the difference between the current position orientation (x, y, θ) and the desired position orientation (x_d, y_d, θ_d) of a given trajectory. The whole control scheme is shown in figure 12.

7 Conclusions

In this work, we tried to replicate analogous results to those obtained in [PBC17] and [APA+19]. We started from the classical model of the WIP whose derivation is depicted in section 2. Even if the main objective of this work was the reproduction of the experiments shown in [PBC17] and [APA+19], we didn't stop there.

Since all of us were relatively new about the Variational Integrator, we spent some time to understand this new instrument, this leaded to the development of the section 3, in which we gave a more general description of this appealing integration technique and we experimented with a simple and well known classic case of study: the discretization of a pendulum dynamics. In these simulations we compared this integration technique with the more common Runge-Kutta method showing the

superior performances of the former, mainly from the energy conservation point of view.

As done in the original papers, from the application of the Variational Integrator we derived in section 4 an implicit discrete-time model of the WIP that we have then exploited to design optimal trajectories through the use of the IPOPT solver, reproducing simulations in the same context of [APA+19]. However we pushed further the line by developing an explicit discrete-time model of the WIP. This last model provided us some advantage with respect to the implicit model, indeed, as we expected, the computational time for the optimization procedure was much less and furthermore we were able to directly simulate the WIP model without resorting to the use of IPOPT.

Finally, in the last section we developed a LQR controller for the WIP. This controller is added in feedback mode in a dual control scheme to mitigate unaccounted disturbances and to guarantee that the optimal trajectory, given as feedforward, is robustly followed by the WIP.

Appendix

We start by considering the last 2 rows of the matricial equation (15)

$$\Psi(k+1)v_\alpha + \gamma(k+1)v_{\phi_1} + \mathbb{K}(k+1)v_{\phi_2} = \Psi(k)\bar{v}_\alpha + \gamma(k)\bar{v}_{\phi_1} + \mathbb{K}(k)\bar{v}_{\phi_2} + h\tau_1 \tag{22}$$

$$\Psi(k+1)v_\alpha + \mathbb{K}(k+1)v_{\phi_1} + \gamma(k+1)v_{\phi_2} = \Psi(k)\bar{v}_\alpha + \mathbb{K}(k)\bar{v}_{\phi_1} + \gamma(k)\bar{v}_{\phi_2} + h\tau_2 \tag{23}$$

where

$$\Psi(k) = m_b r_w b \cos\alpha_k$$
$$\Phi(k) = m_b r_w b \sin\alpha_k$$
$$\gamma(k) = \mathbb{H}(\alpha_k) + I_{W_{yy}}$$

We will also use $v_i = v_i(k)$, $\bar{v}_i = v_i(k-1)$ to ease the notation.
We observe that, for any k,

$$\gamma(k) - \mathbb{K}(k) = \frac{I_\theta(\alpha_k)}{d_w^2} r_w^2 + I_{W_{yy}} = \xi(k) \tag{24}$$

subtracting 22 to 23

$$\mathbb{K}(k+1)v_{\phi_1} - \gamma(k+1)v_{\phi_1} + \gamma(k+1)v_{\phi_2} - \mathbb{K}(k+1)v_{\phi_2} = \mathbb{K}(k)\bar{v}_{\phi_1} - \gamma(k)\bar{v}_{\phi_1} + \gamma(k)\bar{v}_{\phi_2} - \mathbb{K}(k)\bar{v}_{\phi_2} + h\tau_2 - h\tau_1$$

hence

$$(\gamma(k+1) - \mathbb{K}(k+1))v_{\phi_2} - (\gamma(k+1) - \mathbb{K}(k+1))v_{\phi_1} = (\gamma(k) - \mathbb{K}(k))\bar{v}_{\phi_2} - (\gamma(k) - \mathbb{K}(k))\bar{v}_{\phi_1} + h\tau_2 - h\tau_1$$

and finally, using 24

$$\xi(k+1)(v_{\phi_2} - v_{\phi_1}) = \xi(k)(\bar{v}_{\phi_2} - \bar{v}_{\phi_1}) + h(\tau_2 - \tau_1)$$

this gives us the following result

$$v_{\phi_2} - v_{\phi_1} = (\xi(k)(\bar{v}_{\phi_2} - \bar{v}_{\phi_1}) + h(\tau_2 - \tau_1))\xi(k+1)^{-1} \tag{25}$$

which maps the relative velocity of our wheels to the its next step value. Note that $\xi(k)$ is generally invertible for any k if $I_{W_{yy}} \neq 0$.

Now, we observe that for any k

$$\gamma(k) + \mathbb{K}(k) = 2r_w^2(m_b + 2m_w) + I_{W_{yy}} = \delta \tag{26}$$

Adding 22 and 23

$$2\Psi(k+1)v_\alpha + v_{\phi_1}(\gamma(k+1)+\mathbb{K}(k+1)) + v_{\phi_2}(\gamma(k+1)+\mathbb{K}(k+1)) = 2\Psi(k)\bar{v}_\alpha + \bar{v}_{\phi_1}(\gamma(k)+\mathbb{K}(k)) + \bar{v}_{\phi_2}(\gamma(k)+\mathbb{K}(k)) + h(\tau_1+\tau_2)$$

using 26 we get

$$\delta(v_{\phi_1} + v_{\phi_2}) = 2\Psi(k)\bar{v}_\alpha - 2\Psi(k+1)v_\alpha + \delta(\bar{v}_{\phi_1} + \bar{v}_{\phi_2}) + h(\tau_1 + \tau_2)$$

and therefore

$$v_{\phi_1} + v_{\phi_2} = 2\frac{\Psi(k)\bar{v}_\alpha - \Psi(k+1)v_\alpha}{\delta} + (\bar{v}_{\phi_1} + \bar{v}_{\phi_2}) + \frac{h}{\delta}(\tau_1 + \tau_2) \tag{27}$$

Now, from 25 and 27 we can compute the map for v_{ϕ_1} and v_{ϕ_2}.
if we name

$$D = v_{\phi_1} - v_{\phi_2} = (\xi(k)(\bar{v}_{\phi_1} - \bar{v}_{\phi_2}) + h(\tau_1 - \tau_2))\xi(k+1)^{-1}$$

we get

$$v_{\phi_1} = \frac{v_{\phi_1} + v_{\phi_2} + D}{2} = \frac{\Psi(k)\bar{v}_\alpha - \Psi(k+1)v_\alpha}{\delta} + \frac{\bar{v}_{\phi_1} + \bar{v}_{\phi_2}}{2} + \frac{h}{2\delta}(\tau_1 + \tau_2) + \frac{D}{2} \tag{28}$$

$$v_{\phi_2} = \frac{v_{\phi_1} + v_{\phi_2} - D}{2} = \frac{\Psi(k)\bar{v}_\alpha - \Psi(k+1)v_\alpha}{\delta} + \frac{\bar{v}_{\phi_1} + \bar{v}_{\phi_2}}{2} + \frac{h}{2\delta}(\tau_1 + \tau_2) - \frac{D}{2} \tag{29}$$

We then just need to find v_α to compute the wheels velocities.

We consider now the last row of the matricial equation (15)

$$\breve{M}(k+1)\mathbf{v}_{k+1} - h\breve{C}(k+1) = \breve{M}(k)\mathbf{v}_k + h\tau_0 = \breve{M}(k)\mathbf{v}_k \tag{30}$$

where $\breve{M}(k)$ and $\breve{C}(k)$ are the last row of, respectively, $M(k)$ and $C(k)$. Now, recall that

$$\begin{aligned} \Psi(k) &= m_b b r_w \cos\alpha_k \\ \Phi(k) &= m_b b r_w \sin\alpha_k \end{aligned} \tag{31}$$

and be Ω defined as

$$\Omega = m_b b^2 + I_{B_{yy}}$$

we therefore can formulate $\breve{M}(k)$ as

$$\breve{M}(k) = \begin{bmatrix} \Omega & \Psi(k) & \Psi(k) \end{bmatrix}$$

We can then rewrite eq. (30) as

$$\Omega v_\alpha + \Psi(k+1)v_{\phi_1} + \Psi(k+1)v_{\phi_2} - h\breve{C}(k+1) = \Omega\bar{v}_\alpha + \Psi(k)\bar{v}_{\phi_1} + \Psi(k)\bar{v}_{\phi_2}$$

If we name the right part of the equation as

$$Z = \Omega\bar{v}_\alpha + \Psi(k)\bar{v}_{\phi_1} + \Psi(k)\bar{v}_{\phi_2}$$

we have

$$\Omega v_\alpha + \Psi(k+1)(v_{\phi_1} + v_{\phi_2}) - h\breve{C}(k+1) = Z$$

and therefore, using (27)

$$\Omega v_\alpha + \Psi(k+1)(2\frac{\Psi(k)\bar{v}_\alpha - \Psi(k+1)v_\alpha}{\delta} + (\bar{v}_{\phi_1} + \bar{v}_{\phi_2}) + \frac{h}{\delta}(\tau_1 + \tau_2)) - h\breve{C}(k+1) = Z \tag{32}$$

Since $\breve{C}(k+1)$ is defined as

$$\breve{C}(k+1) = \frac{r_w^2}{2d_w^2}(I_{B_{xx}} - I_{B_{zz}} + m_b b^2)\sin 2\alpha(v_{\phi_1} - v_{\phi_2})^2 - m_b b r_w \sin\alpha v_\alpha(v_{\phi_2} + v_{\phi_1}) + m_b b g \sin\alpha$$

We can then use (25), (27) and (31) to get

$$\breve{C}(k+1) = \frac{r_w^2}{2d_w^2}(I_{B_{xx}} - I_{B_{zz}} + m_b b^2)\sin 2\alpha D^2 - v_\alpha\Phi(k+1)(\eta - 2\frac{\Psi(k+1)v_\alpha}{\delta}) + \frac{g}{r_w}\Phi(k+1) \tag{33}$$

with

$$\eta = v_{\phi_2} + v_{\phi_1} + 2\frac{\Psi(k+1)v_\alpha}{\delta} = 2\frac{\Psi(k)\bar{v}_\alpha}{\delta} + (\bar{v}_{\phi_1} + \bar{v}_{\phi_2}) + \frac{h}{\delta}(\tau_1 + \tau_2) \tag{34}$$

Our goal now is to use (32) and (33) to get a second degree equation in v_α to be solved.
We start by writing (32) as

$$v_\alpha(\Omega - 2\frac{\Psi(k+1)^2}{\delta}) + \Psi(k+1)(2\frac{\Psi(k)\bar{v}_\alpha}{\delta} + (\bar{v}_{\phi_1} + \bar{v}_{\phi_2}) + \frac{h}{\delta}(\tau_1 + \tau_2)) - h\breve{C}(k+1) = Z$$

and therefore, using (34), we get

$$v_\alpha(\Omega - 2\frac{\Psi(k+1)^2}{\delta}) - h\breve{C}(k+1) = Z - \Psi(k+1)\eta$$

Indeed, using (34) again we can also rewrite (33) as

$$\breve{C}(k+1) = -\Phi(k+1)(-2\frac{\Psi(k+1)v_\alpha}{\delta} + \eta)v_\alpha + C_0 \tag{35}$$

where C_0 is the part of $\breve{C}(k+1)$ which does not depend on v_α.

$$C_0 = \frac{r_w^2}{2d_w^2}(I_{B_{xx}} - I_{B_{zz}} + m_b b^2)\sin 2\alpha_{k+1}D^2 + \frac{g}{r_w}\Phi(k+1)$$

By using (35) in eq.(32) we finally obtain a second degree equation in v_α

$$v_\alpha(\Omega - 2\frac{\Psi(k+1)^2}{\delta}) + h\Phi(k+1)(-2\frac{\Psi(k+1)v_\alpha}{\delta} + \eta)v_\alpha = Z + hC_0 - \Psi(k+1)\eta$$

and, more explicitly

$$2\frac{\Psi(k+1)}{\delta}h\Phi(k+1)v_\alpha^2 + (\Omega - 2\frac{\Psi(k+1)^2}{\delta} + h\Phi(k+1)\eta)v_\alpha = Z + hC_0 - \Psi(k+1)\eta$$

If we define

$$\lambda = \frac{2h}{\delta}\Psi(k+1)\Phi(k+1)$$

$$\rho = \Omega - \frac{2}{\delta}\Psi(k+1)^2 + h\Phi(k+1)\eta$$

$$\mu = Z + hC_0 - \Psi(k+1)\eta$$

we finally obtain

$$\lambda v_\alpha^2 + \rho v_\alpha - \mu = 0$$

Is empirically easy to determine that the solution for v_α is the one defined as

$$v_\alpha = \frac{-\rho + \sqrt{\rho^2 + 4\lambda\mu}}{2\lambda} \tag{36}$$

or, equivalently

$$v_\alpha = \frac{2\mu}{\rho + \sqrt{\rho^2 + 4\lambda\mu}} = \frac{\mu}{\rho}\frac{2}{1 + \sqrt{1 + \frac{4\lambda\mu}{\rho^2}}}$$

which is better for computational reasons since the same equation holds also for $\lambda = 0$ and therefore we can easily vectorialize the code for the *model consistency constraint* (state constraints) evaluation during IPOPT optimization. Notice that for $\lambda \to 0$ there's continuity for the solution. This continuity argument is our strongest driver behind discarding

$$v_\alpha = \frac{-\rho - \sqrt{\rho^2 + 4\lambda\mu}}{2\lambda}$$

which, for $\lambda \to 0$, tends to $\pm\infty$.

References

[APA+19] Klaus Albert, Karmvir Singh Phogat, Felix Anhalt, Ravi N Banavar, Debasish Chatterjee, and Boris Lohmann. Structure-preserving constrained optimal trajectory planning of a wheeled inverted pendulum, 2019.

[CMR01] Hernán Cendra, Jerrold E Marsden, and Tudor S Ratiu. Geometric mechanics, lagrangian reduction, and nonholonomic systems. In *Mathematics unlimited—2001 and beyond*, pages 221–273. Springer, 2001.

[IMKNZ00] Arieh Iserles, Hans Z. Munthe-Kaas, Syvert P. Nørsett, and Antonella Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.

[KM97] Wang-Sang Koon and Jerrold E Marsden. Optimal control for holonomic and nonholonomic mechanical systems with symmetry and lagrangian reduction. *SIAM Journal on Control and Optimization*, 35(3):901–929, 1997.

[KTG16] Michael Kraus, Emanuele Tassi, and Daniela Grasso. Variational integrators for reduced magnetohydrodynamics. *Journal of Computational Physics*, 321:435–458, sep 2016.

[Leo07] Melvin Leok. An overview of lie group variational integrators and their applications to optimal control. In *International conference on scientific computation and differential equations*, page 1. The French National Institute for Research in Computer Science and Control, 2007.

[LMOW04] Adrian Lew, Jerrold Marsden, Michael Ortiz, and Matthew West. An overview of variational integrators. *Finite Element Methods: 1970's and Beyond*, 01 2004.

[Ngu20] Nam Nguyn. Control of two-wheeled inverted pendulum robot using robust pi and lqr controllers. *Journal of Military Science and Technology*, 66A:1–15, 06 2020.

[PBC17] Karmvir Phogat, Ravi Banavar, and Debasish Chatterjee. Structure-preserving discrete-time optimal maneuvers of a wheeled inverted pendulum. *IFAC-PapersOnLine*, 51, 10 2017.

[SK22] Manmohan Sharma and Indrani Kar. Modeling, control and variational integration for an inverted pendulum on s1. 02 2022.