Multibody Syst Dyn (2018) 43:297–319
https://doi.org/10.1007/s11044-018-9626-7

# Validation of multibody modeling and simulation using an instrumented bicycle: from the computer to the road

José L. Escalona[1,2] · Adam Kłodowski[3] ·
Sergio Muñoz[2]

Received: 21 September 2016 / Accepted: 5 April 2018 / Published online: 19 April 2018

**Abstract** This paper describes the use of an instrumented bicycle and its computational model for teaching multibody dynamics. The presented approach employs the Whipple model for the kinematic and inverse dynamic simulation of a bicycle ride using as an input three generalized coordinates registered with digital sensors. During the experimental phase, students ride the instrumented bicycle to collect the necessary sensor data. The kinematic and inverse dynamic simulations based on these signals provide a full picture of the motion of the system in different positions and at a range of velocities and accelerations. In addition, they estimate the traction, control, and tire-to-road contact forces during the ride. To validate the simulated results, the simulated velocity and accelerations are compared with the data acquired with an inertial measurement unit (IMU) installed on the bicycle. The paper describes the experimental setup of the instrumented bicycle, enabling readers to build the very same system for their own educational use. The instrumented bicycle system is based on open-source software and as much as possible on open hardware.

**Keywords** Education of multibody dynamics · Real-time simulator · Vehicle stability · Symbolic computation

## 1 Introduction

This paper is the continuation of a previous one [1] in which the authors describe the use of a bicycle multibody model (Whipple model) to teach multibody dynamics. The model itself and the dynamic properties of the bicycle are ideal tools to design a varied, mod-

✉ S. Muñoz
sergiomunoz@us.es

[1] Department of Engineering, Aarhus University, Aarhus, Denmark

[2] Technical School of Engineering, University of Seville, Seville, Spain

[3] Department of Mechanical Engineering, Lappeenranta University of Technology, Lappeenranta, Finland

ern, and enjoyable multibody course. The course was designed for graduate students, and it deals with the kinematics and dynamics of multibody systems, stability analysis, computer implementation, experimental validation, and real-time interactive simulation. This paper examines the possibilities of the bicycle model to teach multibody dynamics in an undergraduate course. In fact, this paper shows the content of the course "Kinematics and Dynamics of Machines" delivered at the third-year BSc level at the University of Seville from the academic year 2011–2012 to 2015–2016. The response from the students was very positive.

The well-known *Whipple bicycle model* [2], which consists of four rigid bodies with constraints, is an excellent example for teaching multibody dynamics to engineering students. The apparently simple bicycle model contains a large variety of elements that make it very appealing for teaching purposes. Therefore, the model can be used in courses from intermediate to advanced, allowing students to better understand the analytical dynamics of constrained mechanical systems as well as computational techniques applied to a system with which they are familiar. The educational developments in this and the former paper [1] are inspired by the excellent work of Schwab and co-workers [3–5] in the field of bicycle dynamics. Paper [3] is a gentle introduction to the Whipple model and the dynamics of the bicycle that gives the students the opportunity to benchmark their modeling results.

The literature on the education of multibody dynamics includes some works related to appropriate theoretical approaches, the design of courses including student projects, and the description of multibody software for teaching purposes. The work of Frączek and Wojtyra [6] describes the design of Multibody Systems (MBS) courses for undergraduate, graduate, and doctoral students and the relation of an MBS course to other subjects. Cavacece et al. [7] report their experiences in introducing MBS courses in engineering education, focusing also on the course design. Pennestrì and Vita [8] explain the theoretical and numerical methods that are used for kinematic and dynamic analysis and multibody computer programming. Fissete and Samin [9, 10] have applied the concept of *project-based learning* to the education of multibody dynamics. In a more recent work, Docquier et al. [11] have used a mountain bike as an example to describe modeling hypotheses and simplifications in multibody dynamics. García de Jalón and Callejo [12–14] describe a methodology for teaching multibody dynamics based on the use of *natural coordinates*. Petuya et al. [15] describe educational software for teaching the kinematics of spatial mechanisms. Rideout [16] presents an intuitive method to teach multibody dynamics based on parasitic elements (stiff springs to allow a small but finite relaxation of ideal joint constraints). Wolfsteiner [17] has proposed a methodology for learning multibody dynamics with Matlab based on the symbolic derivation of equations and their numerical solution.

This paper is a conceptually different educational approach to multibody dynamics. In the proposed course, it is assumed that students already have a background in the kinematics and dynamics of machines and analytical mechanics. The bicycle example is then used to fix the concepts in an *example-oriented course* dealing with the kinematics and dynamics of multibody systems, computer implementation and experimental validation. This paper is the result of four years of teaching multibody dynamics to undergraduate students at the University of Seville using the bicycle as a theme. The course includes three lessons. Each lesson consists of three parts: theory and a related example, and one experimental activity in the laboratory, as follows:

*Lesson 1*

**Theory:** Coordinate selection, frames and constraints in multibody systems.
**Example:** Whipple bicycle model.
**Experiment:** Experiment with instrumented bicycle and signal processing.

*Lesson 2*

**Theory:** Kinematics of multibody systems.
**Example:** Simulation of a bicycle; treatment of non-holonomic constraints.
**Experiment:** Simulation of the bicycle driven by experimental measurements (sensor data); experimental validation of the multibody kinematic model.

*Lesson 3*

**Theory:** Dynamics of multibody systems.
**Example:** Inverse dynamics of the bicycle; calculation of reaction, control and drive generalized forces.
**Experiment:** Inverse dynamics of the bicycle driven by experimental measurements; calculation of rider actions on the bicycle and wheel–road contact forces during the ride.

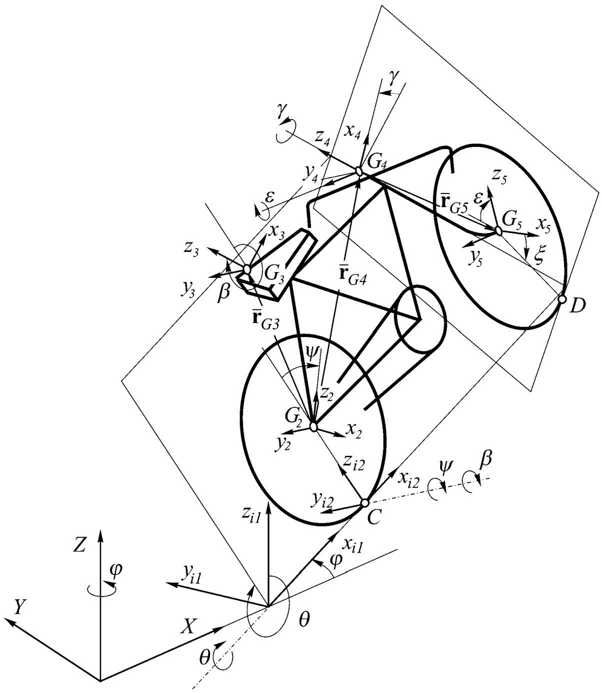
This paper is organized as follows: the next section briefly describes the basic assumptions in the model, generalized coordinates and frames. It also includes a description of the symbolic calculation of velocities and accelerations and the kinematic constraints and their derivatives. Section 3 is devoted to the kinematic simulation of the bicycle model based on experimentally measured data. Section 4 establishes the symbolic procedure for computing the equations of motion of the bicycle model and the assumptions made about the applied forces. Section 5 proposes a computational procedure for the inverse dynamic analysis of the bicycle model based on experimentally measured data. Section 6 describes the full experimental setup. Sensors, the microcontroller, computers, connections, communications, software and programming languages are detailed. Section 7 shows the proposed method to compare the computer simulations based on multibody dynamics with the experimentally measured data. Finally, Sect. 8 shows these comparisons and the interpretation of the results with an educational perspective.

## 2 Multibody model of the bicycle

The multibody model of the bicycle was explained in detail in [1]. This section provides a summary of the model. Coordinates used in the multibody formulations can be divided into reference coordinates that are associated with each body in the system and relative or joint coordinates. The former coordinate family results in highly constrained formulations, easy to implement in general purpose computer programs. The latter family results in less constrained and more efficient formulations requiring more user involvement in the modeling process. This paper selects a set of relative coordinates to model the bicycle, assuming it to be a closed-loop single-chain mechanism. This selection results in a minimal set of coordinates.

Figure 1 shows the bicycle in an arbitrary position, and the frames and coordinates used. The model includes four moving bodies: rear wheel (body 2), rear frame-rider (body 3), steering assembly (body 4) and front wheel (body 5). The wheels are modeled as circles (without volume). The road is assumed planar and horizontal. Rolling without slipping of

**Fig. 1** Coordinates and frames for kinematic description



the wheels is assumed. The body frames are located at the center of the mass of each moving body. The selected relative coordinates are grouped in the following column matrix:

$$\boldsymbol{q} = \begin{bmatrix} x_C & y_C & \phi & \theta & \psi & \beta & \gamma & \varepsilon & \xi \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

where

- coordinates $x_C$ and $y_C$ are the Cartesian components of the position vector of the contact point $C$ of the rear wheel in the global plane $\langle XY \rangle$;
- angle $\varphi$ is the heading angle (yaw);
- angle $\theta$ is the lean angle (roll);
- angle $\psi$ is the rolling angle of the rear wheel (pitch);
- angle $\beta$ is the pitch angle of the rear frame;
- angle $\gamma$ is the steering angle;
- angle $\epsilon$ is the rolling angle of the front wheel (pitch);
- angle $\xi$ is the angle of the radius that contains the contact point of the front wheel in the body frame.

The selected coordinates enable the calculation of the position vector of the origin and orientation matrix of the body frames with respect to the global frame as follows:

$$\boldsymbol{R}^i = \boldsymbol{R}^i(\boldsymbol{q}), \qquad \boldsymbol{A}^i = \boldsymbol{A}^i(\boldsymbol{q}), \quad i = 2, 3, 4, 5. \tag{2}$$

The absolute velocity vector and angular velocity vectors are computed using Jacobian matrices $\boldsymbol{H}^i$ and $\bar{\boldsymbol{G}}^i$ as follows:

$$\boldsymbol{v}_G^i = \boldsymbol{H}^i(\boldsymbol{q})\dot{\boldsymbol{q}}, \qquad \bar{\boldsymbol{\omega}}^i = \bar{\boldsymbol{G}}^i(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad i = 2, 3, 4, 5, \tag{3}$$

where the bar over the symbol means that the components of the vector are given in the body frame. The Jacobian matrices can be obtained symbolically from the expressions given in (2) as follows:

$$\begin{aligned} \boldsymbol{H}^i &= \frac{\partial \boldsymbol{R}^i}{\partial \boldsymbol{q}}, \qquad \bar{\boldsymbol{G}}^i = \frac{\partial \bar{\boldsymbol{\omega}}^i}{\partial \dot{\boldsymbol{q}}} \\ \bar{\boldsymbol{\omega}}^i &= \operatorname{axial}\left(\boldsymbol{A}^{i\mathrm{T}} \dot{\boldsymbol{A}}^i\right), \quad i = 2,3,4,5. \end{aligned} \tag{4}$$

Similarly, the acceleration and angular acceleration of the moving bodies are:

$$\boldsymbol{a}_G^i = \boldsymbol{H}^i(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}^i(\boldsymbol{q})\dot{\boldsymbol{q}}, \qquad \bar{\boldsymbol{\alpha}}^i = \bar{\boldsymbol{G}}^i(\boldsymbol{q})\ddot{\boldsymbol{q}} + \bar{\boldsymbol{g}}^i(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad i = 2,3,4,5, \tag{5}$$

where also the new Jacobian matrices in this expression are obtained symbolically as follows:

$$\boldsymbol{h}^i = \frac{\partial \boldsymbol{v}_G^i}{\partial \boldsymbol{q}}, \qquad \bar{\boldsymbol{g}}^i = \frac{\partial \bar{\boldsymbol{\omega}}^i}{\partial \boldsymbol{q}}, \quad i = 2,3,4,5. \tag{6}$$

Note that once the relations given in Eq. (2) are established, the velocities and accelerations of the moving bodies can be calculated automatically and systematically using symbolic computation.

The coordinates in Eq. (1) are not free but subjected to six nonlinear constraints as follows:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}^{\mathrm{con}}(\boldsymbol{q}) \\ \boldsymbol{C}^{\mathrm{rws}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} = \boldsymbol{0}, \tag{7}$$

where $\boldsymbol{C}^{\mathrm{con}}$ refers to the two holonomic contact constraints that guarantee that the front wheel has a contact point on the road and $\boldsymbol{C}^{\mathrm{rws}}$ stands for the four non-holonomic rolling-without-slipping constraints (two for each wheel). The $\boldsymbol{C}^{\mathrm{rws}}$ constraints are linear in the generalized velocities. The equations can be written as:

$$\boldsymbol{C}^{\mathrm{rws}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{B}^{\mathrm{rws}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{8}$$

where $\boldsymbol{B}^{\mathrm{rws}}$ is the coordinate-dependent Jacobian of the constraints with respect to the generalized coordinates and can be computed symbolically. Details about the definition and symbolic computation of these constraints can be found in [1]. Since the multibody model of the bicycle used $n = 9$ generalized coordinates subjected to $m = 6$ constraints, the model has $g = n - m = 3$ degrees of freedom.

## 3 Kinematic simulation of the bicycle

### 3.1 Kinematic simulation of multibody systems with holonomic constraints

Given a multibody system kinematically described with $n$ generalized coordinates $\boldsymbol{q}$ subjected to $m$ constraints $\boldsymbol{C}$, find $(n - m)$ mobility constraints $\boldsymbol{C}^{\mathrm{mob}}$ (rheonomic constraints, explicit functions of time) defined in a time interval $t_{\mathrm{span}} = \{t_0\ t_{\mathrm{end}}\}$ and calculate the numerical value of the generalized coordinates $\boldsymbol{q}(t_i)$ (position problem), the generalized velocities $\dot{\boldsymbol{q}}(t_i)$ (velocity problem) and generalized accelerations $\ddot{\boldsymbol{q}}(t_i)$ (acceleration problem) in a set of time instants $t_i \in t_{\mathrm{span}}$.

The kinematic simulation of multibody systems with holonomic constraints follows this algorithm:

1. Discretize the time interval into $N$ instants using a small time step $\Delta t$:

$$t_{\text{span}} = \begin{bmatrix} t_0 & t_0 + \Delta t & t_0 + 2 \times \Delta t & \cdots & t_{\text{end}} \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 & \cdots & t_{N-1} \end{bmatrix}.$$

2. For each time instant $t_i$, solve the following:
   (a) Position problem. Calculate the value of the $n$ coordinates solving the $m$ constraint equations $\boldsymbol{C}$ augmented with the $(n-m)$ mobility constraints $\boldsymbol{C}^{\text{mob}}$ using an iterative procedure (Newton–Raphson):

$$\begin{bmatrix} \boldsymbol{C}(\boldsymbol{q}) \\ \boldsymbol{C}^{\text{mob}}(\boldsymbol{q}, t_i) \end{bmatrix} = \boldsymbol{0} \implies \boldsymbol{q}(t_i).$$

   (b) Velocity problem. Calculate the value of the $n$ generalized velocities solving the time derivative of the $m$ constraint equations $\dot{\boldsymbol{C}}$ augmented with the time derivative of the $(n-m)$ mobility constraints $\dot{\boldsymbol{C}}^{\text{mob}}$:

$$\begin{bmatrix} \dot{\boldsymbol{C}}(\boldsymbol{q}) \\ \dot{\boldsymbol{C}}^{\text{mob}}(\boldsymbol{q}, t_i) \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{\boldsymbol{q}} \\ \boldsymbol{C}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix} \dot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{C}_t^{\text{mob}} \end{bmatrix} = \boldsymbol{0} \implies \dot{\boldsymbol{q}}(t_i) = -\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{q}} \\ \boldsymbol{C}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{C}_t^{\text{mob}} \end{bmatrix}.$$

   (c) Acceleration problem. Calculate the value of the $n$ generalized acceleration solving the second time derivative of the $m$ constraint equations $\ddot{\boldsymbol{C}}$ augmented with the second time derivative of the $(n-m)$ mobility constraints $\ddot{\boldsymbol{C}}^{\text{mob}}$:

$$\begin{bmatrix} \ddot{\boldsymbol{C}}(\boldsymbol{q}) \\ \ddot{\boldsymbol{C}}^{\text{mob}}(\boldsymbol{q}, t_i) \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{\boldsymbol{q}} \\ \boldsymbol{C}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \dot{\boldsymbol{C}}_{\boldsymbol{q}} \\ \dot{\boldsymbol{C}}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix} \dot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{0} \\ \dot{\boldsymbol{C}}_t^{\text{mob}} \end{bmatrix} = \boldsymbol{0}$$

$$\implies \ddot{\boldsymbol{q}}(t_i) = -\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{q}} \\ \boldsymbol{C}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix}^{-1} \left( \begin{bmatrix} \dot{\boldsymbol{C}}_{\boldsymbol{q}} \\ \dot{\boldsymbol{C}}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix} \dot{\boldsymbol{q}}(t_i) + \begin{bmatrix} \boldsymbol{0} \\ \dot{\boldsymbol{C}}_t^{\text{mob}} \end{bmatrix} \right).$$

The kinematic simulation of multibody systems with holonomic and non-holonomic constraints, such as the bicycle model, cannot follow this scheme. The reason is that non-holonomic constraints cannot be used in the position problem to obtain the value of the generalized coordinates because these equations also depend on the value of the generalized velocities, which are unknown during this first phase. The only solution to this problem is to perform a numerical time-integration of a subset of the generalized coordinates to obtain their value. Once the position problem is solved, the velocity and acceleration problems follow the procedure described above.

### 3.2 Kinematic simulation of the bicycle based on experimental signals

In a general multibody system, the number of constraints $m$ is the sum of the number of holonomic constraints $m_h$ and the number of non-holonomic constraints $m_{nh}$ $(m = m_h + m_{nh})$. As discussed in the previous subsection, when non-holonomic constraints are used, coordinates are obtained at position level using three different methods: applying mobility constraints to prescribed coordinates, solving holonomic constraints, or performing time-integration. Accordingly, the set of generalized coordinates is divided into three groups:

1. Dynamic coordinates $\boldsymbol{q}^{\text{dyn}}$. The number of dynamic coordinates is equal to the number of degrees of freedom $g = n - m$. These are the coordinates whose time-evolution is prescribed by mobility constraints.
2. Kinematic coordinates $\boldsymbol{q}^{\text{kin}}$. The number of kinematic coordinates is equal to the number of non-holonomic constraints $m_{nh}$. Values of these coordinates are calculated using time-integration.
3. Remaining coordinates $\boldsymbol{q}^{\text{rem}}$. The number of remaining coordinates is equal to the number of holonomic constraints $m_h$. These are the coordinates whose value is calculated by solving the holonomic constraints.

The mathematical conditions for an adequate coordinate partition are explained at the end of this section. In the case of the bicycle, the coordinates are divided as follows:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}^{\text{dyn}} \\ \boldsymbol{q}^{\text{kin}} \\ \boldsymbol{q}^{\text{rem}} \end{bmatrix}, \qquad \boldsymbol{q}^{\text{dyn}} = \begin{bmatrix} \theta \\ \psi \\ \gamma \end{bmatrix}, \qquad \boldsymbol{q}^{\text{kin}} = \begin{bmatrix} x_C \\ y_C \\ \phi \\ \varepsilon \end{bmatrix}, \qquad \boldsymbol{q}^{\text{rem}} = \begin{bmatrix} \beta \\ \xi \end{bmatrix}. \tag{9}$$

This coordinate partitioning follows the modeling experience and fulfills the required mathematical conditions.

In the experiments proposed in this paper, a bicycle is instrumented with a set of sensors that provide the values (signals) of the dynamic coordinates during the experimental ride. Thus, the time-evolution of the dynamic coordinates is described during the simulation according to the experimental data. Sensor signals are used in the mobility constraints of the kinematic simulation of the bicycle as follows:

$$\boldsymbol{C}^{\text{mob}}(\boldsymbol{q}, t) = \boldsymbol{q}^{\text{dyn}} - \begin{bmatrix} \theta^{\text{exp}}(t) \\ \psi^{\text{exp}}(t) \\ \gamma^{\text{exp}}(t) \end{bmatrix} = \boldsymbol{0}, \tag{10}$$

where $\theta^{\text{exp}}$, $\psi^{\text{exp}}$ and $\gamma^{\text{exp}}$ are the experimentally measured lean angle, rear wheel rolling angle and steering angle, respectively. In reality, these are not continuous functions of time but discrete signals. In the kinematic simulation of the bicycle, the time step $\Delta t$ coincides with the sampling period $T_s$ used by the data acquisition system.

The generalized coordinates of the bicycle are subjected to two holonomic contact constraints and four non-holonomic rolling-without-slipping constraints given in Eq. (7), as well as the three rheonomic and holonomic experimental-measure constraints given in Eq. (10). Overall, there are nine coordinates subjected to nine constraints. However, only the five holonomic constraints can be used at position level.

The time-derivative of the holonomic constraints can be augmented with the non-holonomic constraints as follows:

$$\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{q}}^{\text{con}} \\ \boldsymbol{B}^{\text{rws}} \\ \boldsymbol{C}_{\boldsymbol{q}}^{\text{mob}} \end{bmatrix} \dot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{C}_{t}^{\text{mob}} \end{bmatrix} = \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{E} = \boldsymbol{0}, \tag{11}$$

where $\boldsymbol{C}_{\boldsymbol{q}}^{\text{con}}$ is the Jacobian of the holonomic contact constraints, $\boldsymbol{C}_{\boldsymbol{q}}^{\text{mob}}$ is the Jacobian of the experimental measure constraints (Boolean matrix) and $\boldsymbol{C}_{t}^{\text{mob}}$ is given by

$$\boldsymbol{C}_{t}^{\text{mob}} = - \begin{bmatrix} \dot{\theta}^{\text{exp}}(t) \\ \dot{\psi}^{\text{exp}}(t) \\ \dot{\gamma}^{\text{exp}}(t) \end{bmatrix}. \tag{12}$$

As Eq. (11) shows, the Jacobian matrices and time dependent terms are grouped into matrix $\boldsymbol{D}$, which is coordinate-dependent, and $\boldsymbol{E}$. Equation (11) is a linear system of equations that can be solved each instant to find the value of the generalized velocities.

The goal of the kinematic simulation of the bicycle is to obtain the numerical value of all generalized coordinates, velocities and accelerations during the ride from the measured signals given in Eq. (10). The numerical procedure follows this algorithm:

1. The time interval discretization coincides with the set of instants in which the experimental signals have been obtained:

$$t_{\text{span}} = \begin{bmatrix} 0 & T_s & 2\times T_s & \cdots & (N-1)\times T_s \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 & \cdots & t_{N-1} \end{bmatrix}.$$

2. For each time instant $t_i$, solve the following:
   (a) Position problem:
      (i) Obtain the value of $\boldsymbol{q}^{\text{dyn}}$ from the sensor signals (Eq. (11)).
      (ii) Calculate the value of $\boldsymbol{q}^{\text{kin}}$ using a constant acceleration integration scheme:

$$\boldsymbol{q}^{\text{kin}}(t_i) = \boldsymbol{q}^{\text{kin}}(t_{i-1}) + T_s \times \dot{\boldsymbol{q}}^{\text{kin}}(t_{i-1}) + \frac{1}{2}(T_s)^2 \ddot{\boldsymbol{q}}^{\text{kin}}(t_{i-1}).$$

      (iii) Calculate the value of $\boldsymbol{q}^{\text{rem}}$ by solving the holonomic constraints (Newton–Raphson):

$$\boldsymbol{C}^{\text{con}}\big(\boldsymbol{q}^{\text{dyn}}(t_i), \boldsymbol{q}^{\text{kin}}(t_i), \boldsymbol{q}^{\text{rem}}\big) = \boldsymbol{0} \quad \Longrightarrow \quad \boldsymbol{q}^{\text{rem}}(t_i).$$

   (b) Velocity problem:
      Solve Eq. (11) as follows:

$$\dot{\boldsymbol{q}} = -\boldsymbol{D}^{-1}\boldsymbol{E}.$$

   (c) Acceleration problem:
      Solve the time derivative of Eq. (11) as follows:

$$\ddot{\boldsymbol{q}} = -\boldsymbol{D}^{-1}(\dot{D}\dot{q} + \dot{\boldsymbol{E}}).$$

To carry out this kinematic simulation of the bicycle, the following set of arrays and matrices associated with the bicycle constraints have to be computed: $\boldsymbol{C}^{\text{con}}$, $\boldsymbol{B}^{\text{rws}}$, $\boldsymbol{C}_{\boldsymbol{p}}^{\text{con}}$, $\dot{\boldsymbol{C}}_{\boldsymbol{p}}^{\text{con}}$ and $\dot{\boldsymbol{B}}^{\text{rws}}$. These terms are computed symbolically and evaluated numerically at each time step.

To calculate the terms $\boldsymbol{C}_t^{\text{mob}}$ and $\dot{\boldsymbol{C}}_t^{\text{mob}}$ used in the velocity and acceleration problems, the measured signals need to be differentiated. Centered finite differences are used in this work. The differentiation takes place in a pre-processing stage before the kinematic simulation loop. Numerical differentiation amplifies noise and requires numerical filtering. To minimize the effect, digital signals are low-pass filtered using a second order Butterworth with a 5 Hz cut-off frequency. To avoid phase delay, the filter is applied twice, once in the forward and once in the backward direction of time. Second order differentiation of the digital signals is conducted through the first order differentiation of the first numerical derivative instead of the second order numerical differentiation of the original signal. Experience shows that this method produces smoother results. The first order derivatives are filtered twice more before the second order derivatives are calculated.

To start the kinematic simulation process, it is assumed that in the initial instant $t_1$, the dynamic and kinematic coordinates have zero value (the rear wheel is assumed to be over the origin of the global frame pointing in the $X$ direction), and all generalized velocities and accelerations are zero in this and the previous instants.

Going back to the problem of the coordinate partitioning, the different sets of coordinates must fulfill the following conditions for the kinematic simulation to be solvable:

1. Dynamic coordinates must be such that the coefficient matrix $\boldsymbol{D}^{\text{kinrem}}$ used in the velocity and acceleration analyzes is non-singular at each time step.
2. The remaining coordinates must be such that the Jacobian of $\boldsymbol{C}^{\text{con}}$ with respect to them is not singular at each time step. This condition allows calculating their value using the Newton–Raphson method once the value of the dynamic and kinematic coordinates is known.

## 4 Equations of motion of the bicycle

### 4.1 Symbolic calculation of equations of motion

The procedure used in this paper to obtain the equations of motion of the bicycle starts with the assembly of the Newton–Euler equations of the moving rigid bodies. These equations are transformed into the bicycle equations of motion using the kinematic relations given in Eqs. (4)–(5). Assuming the body frames to be attached to the centers of gravity, the Newton–Euler equations of the moving bodies are given by

$$\begin{bmatrix} m^i\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{I}}^i \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_G^i \\ \bar{\boldsymbol{\alpha}}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^i \\ \bar{\boldsymbol{M}}^i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\bar{\boldsymbol{\omega}}^i \wedge \bar{\boldsymbol{I}}^i \bar{\boldsymbol{\omega}}^i \end{bmatrix}, \quad i = 2, 3, 4 \text{ and } 5, \tag{13}$$

where $\mathbf{1}$ is the $3 \times 3$ identity matrix, $m^i$ is the mass, $\bar{\boldsymbol{I}}^i$ are the components of the inertia tensor in the body frame, $\boldsymbol{F}^i$ and$\bar{\boldsymbol{M}}^i$ are the resultant of the forces and moments acting on the center of gravity, and the bar over the symbols means that the components are given in the body frame. The assembly of the Newton–Euler equations of the four moving bodies results in the following:

$$\underbrace{\begin{bmatrix} m^2\mathbf{1} & & & & & \\ & \ddots & & & & \\ & & m^5\mathbf{1} & & & \\ & & & \bar{\boldsymbol{I}}^2 & & \\ & & & & \ddots & \\ & & & & & \bar{\boldsymbol{I}}^6 \end{bmatrix}}_{\hat{\boldsymbol{M}}} \begin{bmatrix} \boldsymbol{a}_G^2 \\ \vdots \\ \boldsymbol{a}_G^5 \\ \bar{\boldsymbol{\alpha}}^2 \\ \vdots \\ \bar{\boldsymbol{\alpha}}^5 \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{F}^2 \\ \vdots \\ \boldsymbol{F}^5 \\ \bar{\boldsymbol{M}}^2 \\ \vdots \\ \bar{\boldsymbol{M}}^5 \end{bmatrix}}_{\hat{\boldsymbol{Q}}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\bar{\boldsymbol{\omega}}^2 \wedge \bar{\boldsymbol{I}}^2 \bar{\boldsymbol{\omega}}^2 \\ \vdots \\ -\bar{\boldsymbol{\omega}}^5 \wedge \bar{\boldsymbol{I}}^5 \bar{\boldsymbol{\omega}}^5 \end{bmatrix}}_{\hat{\boldsymbol{Q}}_v}. \tag{14}$$

Using Eqs. (4)–(5), the accelerations can be written as

$$\begin{bmatrix} \boldsymbol{a}_G^2 \\ \vdots \\ \boldsymbol{a}_G^5 \\ \bar{\boldsymbol{\alpha}}^2 \\ \vdots \\ \bar{\boldsymbol{\alpha}}^5 \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{H}^2 \\ \vdots \\ \boldsymbol{H}^5 \\ \bar{\boldsymbol{G}}^2 \\ \vdots \\ \bar{\boldsymbol{G}}^5 \end{bmatrix}}_{\boldsymbol{L}} \ddot{\boldsymbol{q}} + \underbrace{\begin{bmatrix} \boldsymbol{h}^2 \\ \vdots \\ \boldsymbol{h}^5 \\ \bar{\boldsymbol{g}}^2 \\ \vdots \\ \bar{\boldsymbol{g}}^5 \end{bmatrix}}_{\boldsymbol{l}} \dot{\boldsymbol{q}}. \tag{15}$$

Substituting Eq. (15) into Eq. (14) yields

$$\hat{\boldsymbol{M}}(\boldsymbol{L}\ddot{\boldsymbol{q}} + \boldsymbol{l}\dot{\boldsymbol{q}}) = \hat{\boldsymbol{Q}} + \hat{\boldsymbol{Q}}_v. \tag{16}$$

Multiplying Eq. (16) by $\mathbf{L}^{\mathrm{T}}$ and reordering yields

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{Q} + \boldsymbol{Q}_v, \tag{17}$$

where

$$\boldsymbol{M} = \boldsymbol{L}^{\mathrm{T}}\hat{\boldsymbol{M}}\boldsymbol{L}, \qquad \boldsymbol{Q} = \boldsymbol{L}^{\mathrm{T}}\hat{\boldsymbol{Q}}, \qquad \boldsymbol{Q}_v = \boldsymbol{L}^{\mathrm{T}}(\hat{\boldsymbol{Q}}_v - \hat{\boldsymbol{M}}\boldsymbol{l}\dot{\boldsymbol{p}}). \tag{18}$$

The vector of generalized forces $\boldsymbol{Q}$ in Eq. (17) includes applied forces $\boldsymbol{Q}_{\mathrm{ap}}$ and reaction forces $\boldsymbol{Q}_{\mathrm{reac}}$. The assumed applied forces $\boldsymbol{Q}_{\mathrm{ap}}$ on the bicycle include gravity forces and aerodynamic resistance. The calculation of all these forces is described next.

### 4.2 Reaction forces

Generalized reaction force vector $\boldsymbol{Q}_{\mathrm{reac}}$ appears due to the constraint equations given in Eq. (7) and the mobility constraints due to the experimental measures given in Eq. (10). This vector is calculated using the Lagrange multiplier technique. Due to the presence of the holonomic and non-holonomic constraints, this vector is given by

$$\boldsymbol{Q}_{\mathrm{reac}} = -\boldsymbol{D}^{\mathrm{T}}\boldsymbol{\alpha}, \tag{19}$$

where $\boldsymbol{D}$, which was defined in Eq. (11), acts as the Jacobian of the constraints in multibody systems with just holonomic constraints, and $\boldsymbol{\lambda}$ is the vector of unknown Lagrange multipliers. One Lagrange multiplier $\lambda_i$ is associated with each constraint. Including Eq. (19) into Eq. (17), one obtains

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}^{\mathrm{T}}\boldsymbol{\alpha} = \boldsymbol{Q}_{\mathrm{ap}} + \boldsymbol{Q}_v. \tag{20}$$

These equations are augmented with the previously calculated second derivative of the constraint equations as follows:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{D}^{\mathrm{T}} \\ \boldsymbol{D} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_{ap} + \boldsymbol{Q}_v \\ -\dot{\boldsymbol{D}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{E}} \end{bmatrix}. \tag{21}$$

### 4.3 Applied forces

The vector of generalized applied forces $\mathbf{Q}_{ap}$ is given by

$$\boldsymbol{Q}_{ap} = \boldsymbol{Q}_{\text{grav}} + \boldsymbol{Q}_{\text{aero}}, \tag{22}$$

where $\boldsymbol{Q}_{\text{grav}}$ is the generalized gravity force and $\boldsymbol{Q}_{\text{aero}}$ is the generalized force due the aerodynamic resistance.

Gravity forces acting on each body are given by

$$\boldsymbol{F}^{i}_{\text{grav}} = \begin{bmatrix} 0 \\ 0 \\ -m^{i} g \end{bmatrix}, \quad i = 2, 3, 4 \text{ and } 5. \tag{23}$$

The total virtual power of gravity forces is given by

$$\dot{W}^{*}_{\text{grav}} = \sum_{i=2}^{5} \left(\boldsymbol{v}^{i\,*}_{G}\right)^{\mathrm{T}} \boldsymbol{F}^{i}_{\text{grav}} = \left(\dot{\boldsymbol{q}}^{*}\right)^{\mathrm{T}} \sum_{i=2}^{5} \boldsymbol{H}^{i^{\mathrm{T}}} \boldsymbol{F}^{i}_{\text{grav}}, \tag{24}$$

where Eq. (3) has been used, and the asterisk refers to virtual magnitudes. Using this equation and the principle of virtual power, vector $\boldsymbol{Q}_{\text{grav}}$ is obtained as

$$\boldsymbol{Q}_{\text{grav}} = \sum_{i=2}^{5} \boldsymbol{H}^{i^{\mathrm{T}}} \boldsymbol{F}^{i}_{\text{grav}}. \tag{25}$$

The aerodynamic resistance is assumed to be proportional to the square of the velocity and to be applied in solid number 3 (rear frame-rider). This is reasonable since this body is clearly the one that generates larger resistance. The modulus of the aerodynamic drag is given by

$$\left|\boldsymbol{F}^{3}_{\text{aero}}\right| = c_v \left(\boldsymbol{v}^{3}_{G}\right)^{\mathrm{T}} \boldsymbol{v}^{3}_{G} = c_v \left|\boldsymbol{v}^{3}_{G}\right|^{2}, \tag{26}$$

where $c_v$ is the aerodynamic coefficient. The direction of this force is opposite to the instantaneous direction of the velocity as follows:

$$\boldsymbol{F}^{3}_{\text{aero}} = -\left|\boldsymbol{F}^{3}_{\text{aero}}\right| \frac{\boldsymbol{v}^{3}_{G}}{\left|\boldsymbol{v}^{3}_{G}\right|} = -c_v \left|\boldsymbol{v}^{3}_{G}\right| \boldsymbol{v}^{3}_{G}. \tag{27}$$

Using again the principle of virtual power, vector $\boldsymbol{Q}_{\text{aero}}$ yields

$$\boldsymbol{Q}_{\text{aero}} = \boldsymbol{H}^{3^{\mathrm{T}}} \boldsymbol{F}^{3}_{\text{aero}} = -c_v \left|\boldsymbol{v}^{3}_{G}\right| \boldsymbol{H}^{3^{\mathrm{T}}} \boldsymbol{v}^{3}_{G}. \tag{28}$$

## 5 Inverse dynamics of the bicycle

In the inverse dynamic analysis of the ride, the objectives are to use the experimentally measured signals to

1. Calculate the actions taken by the cyclist with regard to the bicycle during the ride;
2. Calculate the forces acting between the tires and the road during the ride.

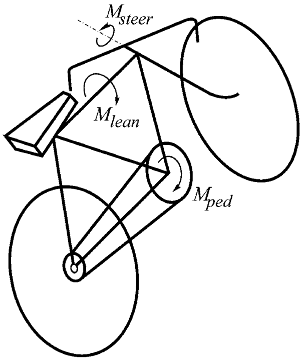


**Fig. 2** Actions of the rider on the bicycle

These generalized forces are calculated from the Lagrange multipliers that appear as unknowns in the linear Eqs. (21). Inverse dynamics simulation is the result of substituting "step c", acceleration analysis, of the previously defined kinematic simulation with the solution of Eq. (21) as follows:

$$\begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{D}^{\mathrm{T}} \\ \boldsymbol{D} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{Q}_{ap} + \boldsymbol{Q}_v \\ -\dot{\boldsymbol{D}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{E}} \end{bmatrix}. \tag{29}$$

Therefore, the inverse dynamics simulation provides the same information as the kinematic simulation combined with the value of the Lagrange multipliers.

As explained in [1], the first contact constraint in $\boldsymbol{C}^{\mathrm{con}}$ guarantees that a point in the front wheel is in contact with the ground while the second one guarantees that the tangent vector to the wheel at the contact point is parallel to the ground. As discussed in [18] the physical meaning of the Lagrange multiplier associated with the first contact constraint $\lambda_1$ is the value of the normal contact force (with opposite sign) in the front wheel and the Lagrange multiplier associated with the second contact constraint $\lambda_2$ is always zero. The four Lagrange multipliers associated with the rolling-without-slipping constraints in $\boldsymbol{C}^{\mathrm{rws}}$ represent the two components of the tangential contact force (with opposite sign) in the rear wheel ($\lambda_3$ and $\lambda_4$) and front wheel ($\lambda_5$ and $\lambda_6$). The physical meaning of the Lagrange multipliers associated with the experimental measures in $\boldsymbol{C}^{\mathrm{mob}}$ represent the generalized forces needed to drive these movements (with opposite sign). Multipliers $\lambda_8$ and $\lambda_9$ associated with the measure of the rear wheel rotation $\psi$ and the steering angle $\gamma$ may be related to the pedaling torque $M_{\mathrm{ped}}$ (or braking torque) and steering torque $M_{\mathrm{steer}}$, respectively, affected by the rate of transmission of the chain in the case of the pedaling torque (see Fig. 2). This interpretation is not precise because these torques are, in reality, internal generalized forces when riding a bicycle. Recall that body 3 combines the rear frame and the rider, and in reality, pedaling and steering are the consequence of internal forces within this compound body. The interpretation of multiplier $\lambda_7$ associated with the measure of the lean angle $\theta$ is more difficult. In fact, there is no lean torque when riding a bicycle. Relative lean between the rider and the frame is a possible way to steer a bicycle, but our model cannot represent this motion and $\lambda_7$ cannot be interpreted as the resulting torque. Therefore, the resulting non-zero multiplier $\lambda_7$ has to be accepted as proof that our model does not accurately represent the riding of a bicycle.

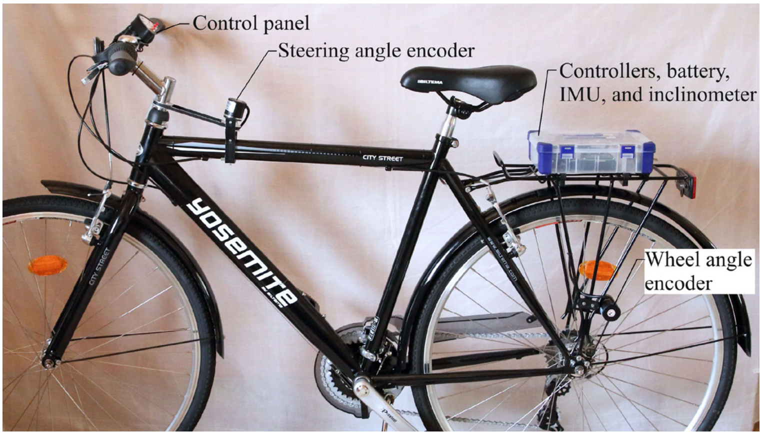


**Fig. 3** Overview of the instrumented bicycle

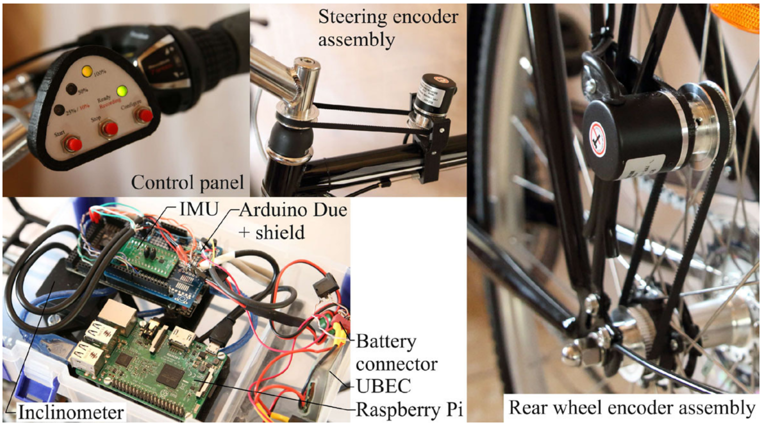


**Fig. 4** Details of instrumented bicycle

## 6 Experimental setup

The experimental setup consists of a bicycle equipped with a data acquisition system powered by a battery. This equipment includes a Raspberry Pi 3 computer, an Arduino Due board and a set of sensors. The data acquisition system communicates via WiFi with the user (“Client”) that processes the data and visualizes the bicycle motion using a laptop or a smart phone. The details of this experimental equipment (hardware and software) are provided in the Appendix including the full component list in Table 1. Figure 3 presents an overview of the system, and Fig. 4 shows detailed photographs of the main components.

The electronic equipment has two main objectives: to record the sensor data that are used as an input to the kinematic (Sect. 3) and inversed dynamic analysis (Sect. 5), and to provide the means for the experimental validation of the kinematic analysis. To this end, the dynamic coordinates $\boldsymbol{q}^{\mathrm{dyn}}$ described previously are measured as follows:

1. The lean angle $\theta$ is measured with an inclinometer.

2. The rolling angle $\psi$ is measured with the rotary encoder installed in the rear wheel and the rear frame.
3. The steering angle $\gamma$ is measured with the rotary encoder installed in the steering assembly and the rear frame.

Because the encoder used to measure the rear wheel angle, $\psi$, is mounted on the rear frame, as can be observed on the right in Fig. 4, the measure is affected by the angle $\beta$ of the rear frame. However, $\psi$ being a large angle of rotation that increases monotonically and $\beta$ a very slightly changing oscillatory angle, the measure of this encoder can be considered as a good estimation of $\psi$.

An *inertial measurement unit* (IMU) including a three-axis accelerometer and a three-axis gyroscope is installed to record the acceleration and angular velocity of the rear frame.

Afterwards, the students use the recorded sensor data to compute offline the kinematic and inverse dynamics of the bicycle using Matlab/Octave. The numerical results are experimentally validated as explained in the next section.

## 7 Validation of kinematic analysis

Validation is important for the students to understand the validity and expected accuracy of the multibody modeling and simulation of the bicycle. Validation is provided at the three levels of the kinematic analysis: position level, velocity level and acceleration level, as shown next.

### 7.1 Validation of position analysis

The output of the position analysis is used to generate a simple computer graphics animation of the bicycle ride using the Matlab/Octave script “AnimateBike.m”, which is given to the students. This computer-generated animation can be shown simultaneously with a video recording of the ride for a full but non-measurable comparison of the position analysis.

Finally, the data provided by the installed IMU can be used in sensor fusion algorithms [19, 20] to determine the orientation of the bicycle rear frame (body 3) during the ride. These algorithms combine information provided by the gyroscope (rate of rotation), the accelerometer (direction of gravity) and in some cases a magnetometer (direction of Earth’s magnetic north) to obtain the actual orientation of the sensor at any instant. This orientation can be compared with the simulated results. Combining the IMU data with GPS data or a computer vision system can also be used to find an accurate estimation of the bicycle trajectory. These solutions have not been implemented in the experimental setup.

### 7.2 Validation of velocity and acceleration analysis

The data acquired by the gyroscope and accelerometer can be used to validate the velocity and acceleration kinematic analysis, respectively. To this end, the angular velocity of the rear frame where the IMU is installed and the acceleration vector of the exact point where this sensor is located have to be evaluated as a function of the computed $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and$\ddot{\boldsymbol{q}}$. The sensor is installed in the rear frame with the sensor-fixed frame parallel to the global frame in the reference position of the bicycle. It is important to consider that the sensor measures the components of the angular velocity and acceleration in the sensor-fixed frame. The gravity force $g$ along the absolute vertical axis $Z$ is a DC signal output in the case of the capacitive accelerometer sensor which is used in the presented setup (piezoelectric and piezoresistive accelerometers behave differently). In the case of the angular velocity, the following formula is used to distinguish the vector components out of the simulated coordinates and velocities:

$$\bar{\boldsymbol{\omega}}_{\mathrm{IMU}} = \boldsymbol{A}_{\beta}\bar{\boldsymbol{\omega}}^{3} = \boldsymbol{A}_{\beta}\bar{\boldsymbol{G}}^{3}\dot{\boldsymbol{q}}, \tag{30}$$

where $\boldsymbol{A}_{\beta}$ is the rotation matrix associated with the angular coordinate $\beta$ and the Jacobian matrix $\bar{\boldsymbol{G}}^{3}$ is defined in Eq. (4). In the case of the acceleration, the following formula is used to derive the vector components out of the simulated coordinates, velocities and accelerations:

$$\bar{\boldsymbol{a}}_{\mathrm{IMU}} = (\boldsymbol{A}_{\phi}\boldsymbol{A}_{\theta})^{\mathrm{T}}\left(\boldsymbol{H}_{\mathrm{IMU}}\ddot{\boldsymbol{q}} + \boldsymbol{h}_{\mathrm{IMU}}\dot{\boldsymbol{q}} + \begin{bmatrix}0 & 0 & g\end{bmatrix}^{\mathrm{T}}\right), \tag{31}$$

where the two first terms in the second set of brackets are the global components of the acceleration vector of the sensor (calculated as in Eq. (5)) and the third term is the acceleration of gravity. These vector components are projected onto the sensor frame using the appropriate transformation matrix. Notice that the rotation due to the frame angle $\beta$ has not been considered in this transformation because its effect is very small. The Jacobian matrices associated with the IMU are symbolically computed as in Eq. (6):

$$\boldsymbol{H}_{\mathrm{IMU}} = \frac{\partial \boldsymbol{R}_{\mathrm{IMU}}^{3}}{\partial \boldsymbol{q}}, \qquad \boldsymbol{h}_{\mathrm{IMU}} = \frac{\partial(\boldsymbol{H}_{\mathrm{IMU}}\dot{\boldsymbol{q}})}{\partial \boldsymbol{q}}. \tag{32}$$

## 8 Experimental simulation results

The experimental results in this section correspond to a 37-second ride with the instrumented bicycle in which a simple closed-loop trajectory is developed: going straight forward approximately 50 m, making a U-turn, going straight 50 m in the opposite direction, making another U-turn and stopping the bicycle in the same position and orientation as when the ride started. During the experiments, students are free to choose the trajectory. The sampling frequency used for all sensors was 100 Hz. The sampling period coincides with the time-step used in the kinematic simulation and inverse dynamic analysis. Therefore, no interpolation of the measured data is needed. Table 1 provides the geometric and inertial parameters and the assumed aerodynamic coefficient. All distances are given in millimeters. Table 1 displays the position of the centers of gravity G2–G5 of the four moving bodies with respect to the center of the rear wheel in the reference configuration. $R_f$ and $R_r$ stand for the radius of the front and rear wheels.

Figure 5 shows some results of the kinematic position analysis. The plot on the left shows a capture of the animation that is generated from the simulation results. The plot on the right of Fig. 5 shows the simulated trace of the contact point in the rear wheel. It can be observed that the simulated trace is not a closed loop, as the contact point does not end in the position where the ride started (at the origin). This is obviously an erroneous result. This problem is due to the accumulation of errors due to the time integration process in the kinematic coordinates $\boldsymbol{q}^{\mathrm{kin}}$ (step 2(a)(ii) of the kinematic analysis). The problem could be resolved by using a more precise integration rule or decreasing the sampling period $T_s$. More precise integration rules, such as multistep or implicit algorithms, are not used in this work because they would complicate in excess the kinematic analysis loop that the students have to program. Decreasing the sampling period is technically difficult.

**Table 1** Parameters of the bicycle

| | Geometric parameters [mm] | | | | |
|---|---|---|---|---|---|
| | G2 | G3 | G4 | G5 | IMU |
| $X$ | 0 | 222 | 863 | 1049 | 60 |
| $Z$ | 0 | 560 | 477 | 0 | 440 |
| | Inertial parameters | | | | |
| | $m$ [kg] | $I_{xx}$ [kg m$^2$] | $I_{yy}$ [kg m$^2$] | $I_{zz}$ [kg m$^2$] | $I_{xz}$ [kg m$^2$] |
| 2 | 2.52 | 0.0884 | 0.1357 | 0.0884 | 0 |
| 3 | 85.0 | 9.2 | 11.0 | 2.8 | −2.4 |
| 4 | 2.96 | 0.3056 | 0.158 | 0.0385 | −0.0385 |
| 5 | 1.96 | 0.0525 | 0.119 | 0.0525 | 0 |
| | Other geometric parameters and aerodynamic coefficients | | | | |
| | $R_f$ | $R_r$ | Head angle [deg] | Wheelbase | $c_v$ [N m$^2$/s$^2$] |
| | 352 | 352 | 19 | 1049 | 0.5 |

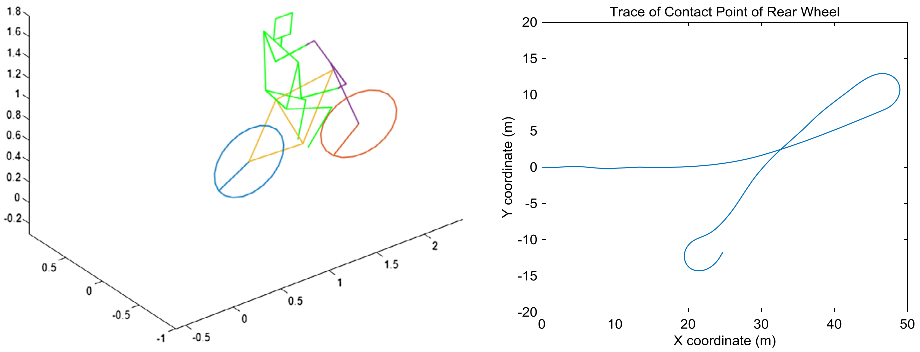


**Fig. 5** Kinematic position analysis. Instantaneous position on the left and simulated trajectory on the right

Nevertheless, the drift problem observed in Fig. 5 cannot be solved completely because all sensors used relative motions of the vehicle for the kinematic simulation measure, thus resulting in a non-observable system. An absolute position signal provided by, for example, GPS is needed for accurate trajectory estimation, as it is well known in vehicle position tracking.

Figure 6 shows the time evolution of the pitch rate ($Y$ component) and yaw rate ($Z$ component) calculated using Eq. (30) with the results of the kinematic velocity analysis (simulation) and compared with the gyroscope measurements (experimental). Very good experimental simulation agreement is found in the yaw rate while very poor agreement is found in the pitch rate. The main reason for the disagreement in the pitch rate is that the bicycle used in the experiments has a front suspension (it was not the bicycle shown in Fig. 3 that is the result of the most recent developments in this work), but the model does not include this additional degree of freedom. The activation of the front suspension allows for a much larger value of the magnitude of the pitch rate. Students learn from this result that the model

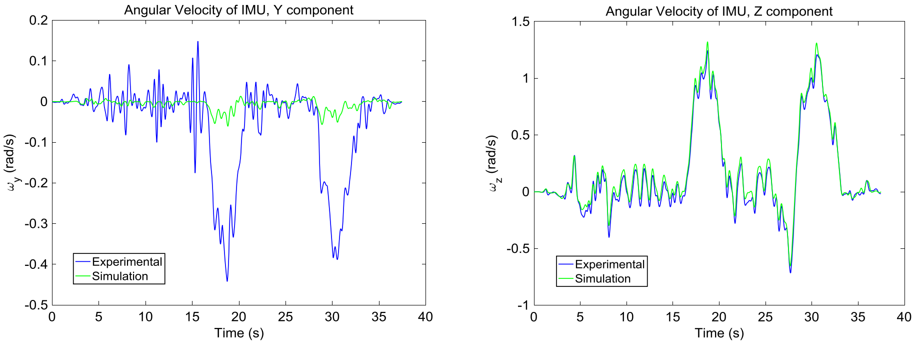


**Fig. 6** Kinematic velocity analysis: simulation vs. experimental. Pitch rate on the left and yaw rate on the right

is just partially valid to describe the vehicle dynamics. The validity of the model depends on its complexity. Of course, this disagreement can be solved by using an instrumented bicycle without front suspension or by including front suspension in the model. However, since other aspects of the vehicle dynamics are well captured, this disagreement is considered as a positive result for educational purposes.

Figure 7 shows the time evolution of the forward acceleration (*X component*) and vertical acceleration (*Z component*) of the point where the IMU is located, calculated using Eq. (31) with the results of the kinematic acceleration analysis ("*Simulation*") and compared with the accelerometer measurements ("*Experimental*"). The forward acceleration shows relatively good agreement, but not as good as the one identified for the yaw rate. As explained before, the acceleration analysis requires the second numerical derivative of the acquired sensor data (encoders and inclinometer) that naturally amplifies the noise in the signals, degrading the quality of the results. On the other hand, also the accelerometer signals contain noise. Better agreement in the forward acceleration can be obtained with proper filtering. However, this is a signal processing exercise going beyond the purpose of this work. In the plot showing the forward acceleration (on the left in Fig. 7), the acceleration and braking periods at the beginning and the end of the ride and the braking before starting the U-turns can be clearly observed. In Fig. 7 and in subsequent plots, the first and second U-turn periods are marked with magenta and cyan vertical lines, respectively. In the vertical direction, the magnitude of the accelerations obtained by simulation are much smaller than those in reality. This effect could be explained by the deformation of the tires during the ride and possibly the influence of the structural flexibility of the bicycle bodies. These possible reasons have not been investigated. However, it is reasonable to conclude again that the model cannot adequately capture this aspect of the vehicle dynamics without increasing its complexity. The vertical acceleration disagreement is also considered as a positive result for educational purposes.

Figures 8 and 9 show some results of the inverse dynamic analysis. The plots in Fig. 8 show the steering torque (left) and pedaling torque (right). Comparing the steering torque, which is obtained as the Lagrange multiplier associated with the steering angle mobility constraint, with the yaw rate of the bicycle (on the right in Fig. 6), one can observe the time evolution of the steering torque during the U-turns, which does not require particularly high but biased values of this torque. The pedaling torque on the right is computed by applying two procedures. The first one is the direct use of the Lagrange multiplier associated with the rear-wheel-angle mobility constraint. The second method is the use of the *reduced inertia concept* employed in machines dynamics. Assuming a simple one-degree-of-freedom planar

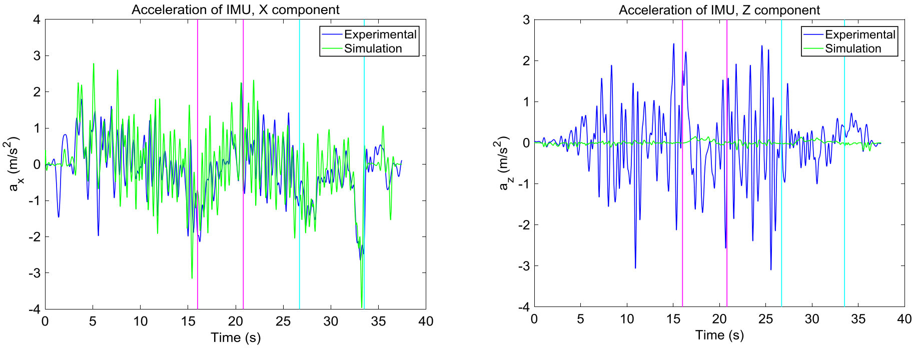


**Fig. 7** Kinematic acceleration analysis: simulation vs. experimental. Forward acceleration on the left and vertical acceleration on the right (Color figure online)

model of the bicycle, the kinetic energy of the system can be computed as a function of the angular velocity of the rear wheel as follows:

$$\left.\begin{aligned} T = \tfrac{1}{2} I_y^2 \dot{\psi}^2 + \tfrac{1}{2} m^2 |\boldsymbol{v}_G^2|^2 + \tfrac{1}{2} m^3 |\boldsymbol{v}_G^3|^2 + \tfrac{1}{2} m^4 |\boldsymbol{v}_G^4|^2 + \tfrac{1}{2} I_y^5 \dot{\varepsilon}^2 + \tfrac{1}{2} m^5 |\boldsymbol{v}_G^5|^2 \\ |\boldsymbol{v}_G^2| = |\boldsymbol{v}_G^3| = |\boldsymbol{v}_G^4| = |\boldsymbol{v}_G^5| = R\dot{\psi}, \quad \dot{\varepsilon} = \dot{\psi} \end{aligned}\right\}$$

$$\Longrightarrow \quad T = \frac{1}{2}\left[I_y^2 + I_y^5 + \left(m^2 + m^3 + m^4 + m^5\right) R^2\right] \dot{\psi}^2 = \frac{1}{2} I_{\text{bicycle}}^2 \dot{\psi}^2 \tag{33}$$

where $R$ is the radius of the rear and front wheels, which are assumed equal, and $I_{\text{bicycle}}^2$ is the moment of inertia of the bicycle reduced to the rear wheel. In the experiment, the value of the reduced moment of inertia is 12.17 kg m$^2$. The green line in Fig. 8 shows the calculation of the pedaling torque as

$$M_{\text{ped}} = I_{\text{bicycle}}^2 \ddot{\psi}, \tag{34}$$

which is obtained from the application of the principle of work and energy. As Fig. 8 demonstrates, both ways to compute the pedaling torque match very well. A difference exists mainly because the motion of the bicycle is three-dimensional and the energy input to the lateral dynamics is not considered in the model used to obtain a reduced inertia.

Figure 9 shows the calculated reaction forces (Lagrange multipliers) in the inverse dynamic analysis. The plot on the left shows the normal contact force in the front wheel ($-\lambda_1$). This force is oscillating around an approximate value of 220 N. Since the total mass of the bicycle and the rider is 93 kg, this load represents approximately one fourth of the total weight of the system being carried by the front wheel during the ride. The plot shows that this force can be doubled during the braking of the bicycle and can be reduced to half due to the vehicle dynamics. This gives an idea of how far the bicycle is from wheel–road separation that would occur if the normal force drops to zero. Because the selected generalized coordinates do not need to be constrained to guarantee the rear wheel contact with the ground [1], the normal contact force on the rear wheel cannot be obtained using an associated Lagrange multiplier. The calculation of the time evolution of this force requires post-processing of the simulation results using equilibrium equations that contain this force as an unknown. The ease of computing the reaction forces is an advantage of the use of reference coordinates instead of the selected minimum set of coordinates.

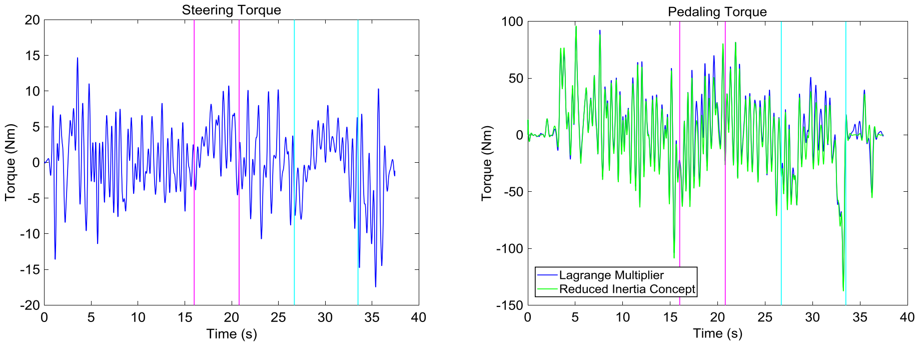


**Fig. 8** Calculated steering torque (left) and pedaling torque (right) during ride (Color figure online)

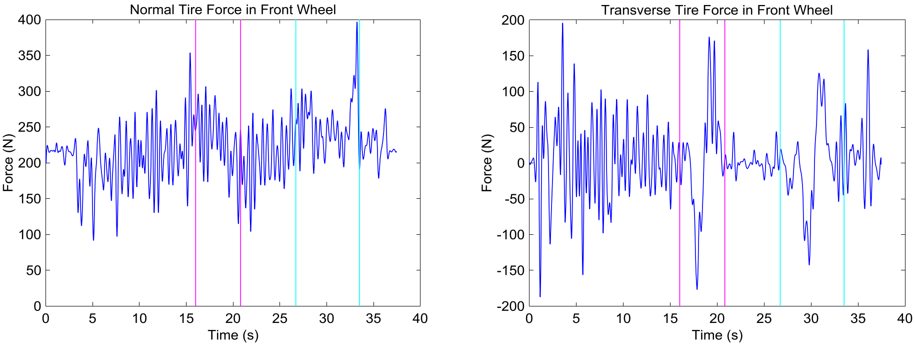


**Fig. 9** Calculated normal tire force (left) and tangential tire force (right) on front wheel during ride

The plot on the right in Fig. 9 shows the transverse tangential contact force in the front wheel. This value is obtained by projecting the total tangential force, whose components are given by $\lambda_5$ and $\lambda_6$ associated with the rolling-without-slipping constraint of the front wheel, in the transverse direction to the front wheel. Projection is necessary because the global components of the absolute tangential velocity of the contact points are used in the rolling-without-slipping constraints as defined in [1]. Therefore, Lagrange multipliers provide the value of the tangential force in the global frame. The time evolution of this transverse contact force during the U-turns can be clearly observed. Figure 10 shows the ratio between the norm of the tangential contact force and the normal contact forces on the front wheel. Peak values occur at the initial instant of acceleration (1.28), the first U-turn (1.06), the second U-turn (0.89) and final braking (0.96). In conclusion, the minimum value of 1.28 of the tire–road coefficient of friction is required to prevent gross sliding of the front wheel during this particular ride. A smaller value of the coefficient of friction would mean that friction could not provide the required force to prevent the wheel from slipping.

## 9 Summary and conclusions

This paper uses the bicycle as an example around which a course on multibody dynamics can be built. In addition, it describes the instrumentation needed to turn a regular bicycle

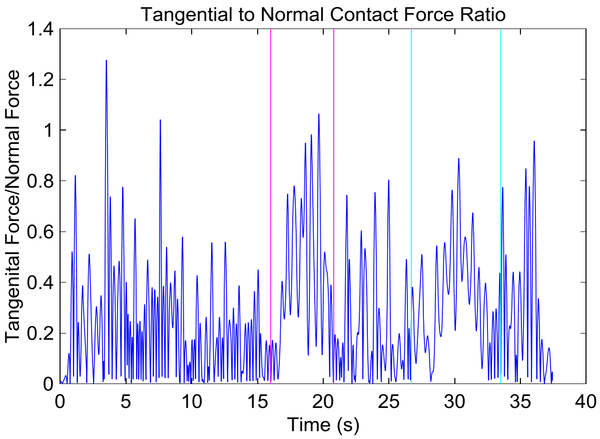


**Fig. 10** Ratio between norm of tangential contact force and normal contact force on front wheel during ride

into an inexpensive and valuable educational instrument in engineering. In fact, the bicycle, its model, and the experiments and computer simulations that can be performed with them can provide to engineering students a wide overview of machine dynamics. Multibody dynamics is considered in this paper as a fundamental discipline with important benefits when combined with others fields: instrumentation, signal processing or Internet communications.

The apparently simple but dynamically rich Whipple model of the bicycle is implemented symbolically to establish the set of kinematic constraint equations and the equations of motion for the kinematic and inverse dynamic simulation of the bicycle, respectively. Based on these equations, algorithms were developed for the kinematic and inverse dynamic simulation of the bicycle with experimentally measured data.

These computer simulations require an instrumented bicycle to collect experimental data. The sensor network, data acquisition system and internet-based communication scheme on the bicycle have been developed in full in this investigation and described in this paper.

Experimental validations of the model and the kinematic simulation have been developed for the position, velocity and acceleration. In the present paper, no validation is included for the inverse dynamic analysis. This is the topic of a future investigation.

Experimental results and their comparison with multibody simulations show very good agreement in some respects and very poor agreement in others. However, the disagreement can be easily explained by the inability of the model to capture some dynamic effects and is therefore considered a positive aspect in the educational sense.

**Acknowledgements** This paper was funded by the Spanish Ministry of Economy and Competitiveness under project reference TRA2014-57609-R. Their support is gratefully acknowledged.

## Appendix: Description of instrumented bicycle

Five sensors are integrated for the measurements: inertial measurement unit (IMU) Bosch BMI160 containing a three-axis digital accelerometer and a three-axis digital gyroscope, a G-NSDOG2-001 two-axis inclinometer from TE Connectivity Sensors and two two-phase incremental rotary encoders LPD3806-600BM-G5-24G. The BMI160 is provided on a shuttle board for easy prototyping. The main specifications of these sensors are:

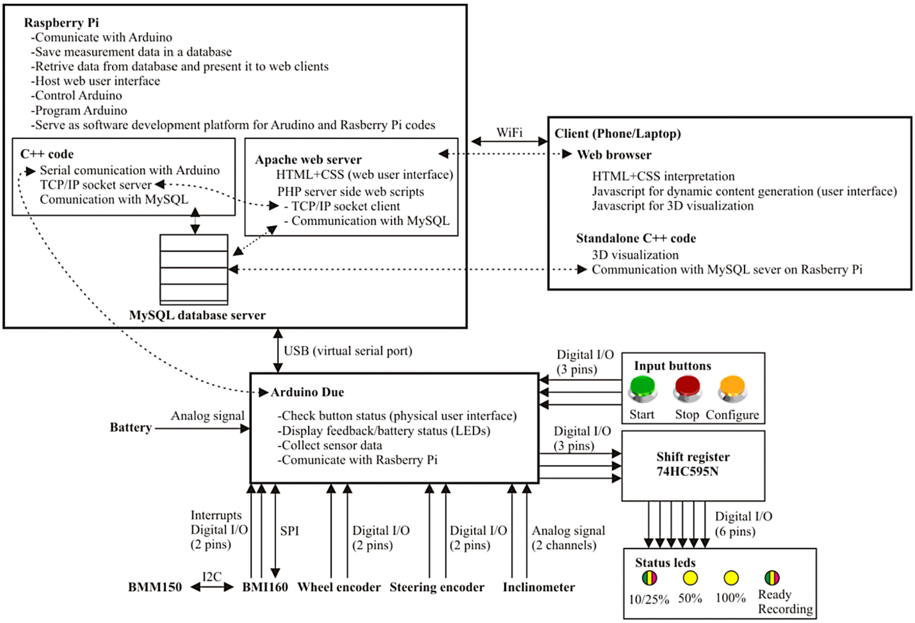


**Fig. 11** Communications and roles of the instrumented bicycle components.

- IMU BMI160: accelerations with a selectable range ±2, ±4, ±8, ±16 g and a 16-bit resolution and an angular velocity of a selectable range ±125, ±250, ±500, ±1000, ±2000°/s and a 16-bit resolution.
- Inclinometer G-NSDOG2-001: Range of ±20° and 12-bit resolution.
- Encoders LPD3806-600BM-G5-24G: 2400 pulses/rev or 0.15° accuracy.

All sensors are connected to the Arduino Due board, which communicates with BMI160 through a four-wire SPI protocol. Rotary encoders are connected to general I/O pins of Arduino, and each encoder requires two pins to determine the rotation direction and number of steps. Raspberry Pi 3 serves as a bridge between low-level time critical data sampling performed by Arduino and high-level processor-intensive tasks of data storage and communication with smart phone and laptops through WiFi. Task divisions between boards and communication protocols are presented in Fig. 11, while electrical connections between components are depicted in Fig. 12. To make the system easier for the user to control, a simple control panel have been designed to provide the basic information—battery charge level, operation status (ready, data acquisition, calibration in progress)—with three buttons which enable the imitation of data acquisition, stopping it, or entering calibration mode (see Fig. 4). The calibration process is performed while keeping the bicycle in the upright positon to set the zero value of the two encoders and the inclinometer and the orientation of the accelerometer with respect to the gravity force.

The software of the system consists of the Arduino Due code, a C++ host program running on Raspberry Pi, and the PHP code on Raspberry Pi for creating a user web interface. All codes are available as supplementary material to this article. To ease data storage and management, the MySQL database engine has been installed on Pi. The web server Apache plus PHP, on the other hand, has been installed for hosting user interface web pages and manuals. In addition, the file server Samba has been installed to facilitate code and data

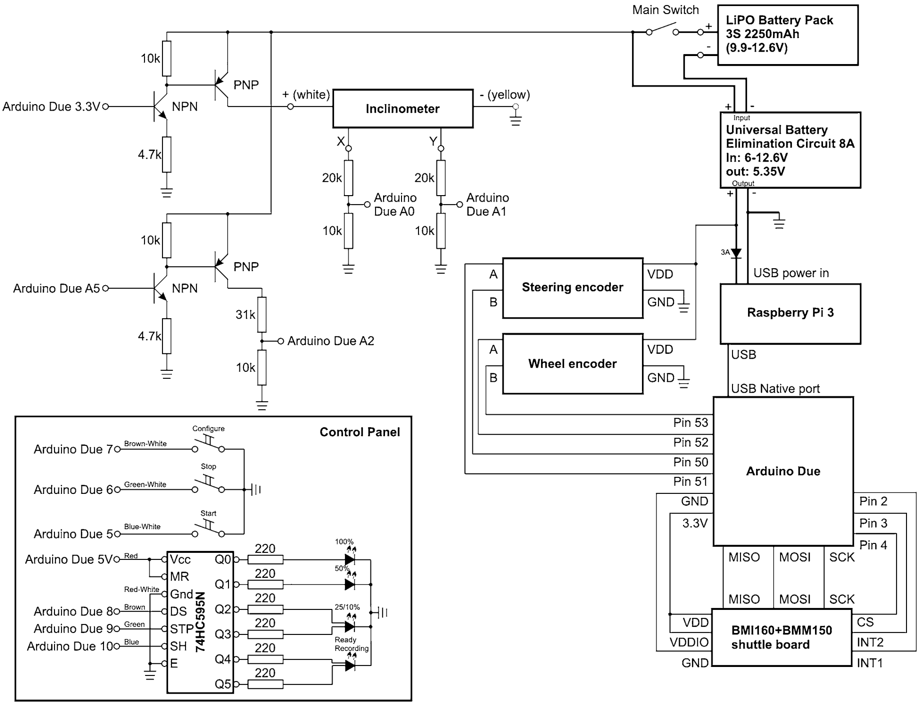


**Fig. 12** Electrical connections between the components of the instrumented bicycle.

transfer during development between Raspberry Pi and a laptop. A tight VNC server has been installed to provide a remote desktop user experience for users connecting from laptops. This eliminates the need of a separate screen, keyboard, and mouse to operate Pi.

Code running on Arduino calibrates all of the sensors, acquiring data from them and transferring the data to Pi. Arduino also handles physical control panel. It is worth noting that this task could equally easily be implemented on Raspberry Pi GPIO pins.

The C++ host program running on Pi is the core of communication between Ardunio and the user. This program acts as a TCP/IP socket server for the PHP code to exchange commands between users running on laptops/phones and Arduino. The TCP/IP socket for communication with Matlab is used to visualize the data and run real-time simulations. Whenever a measurement session is on, the C++ host code collects the data measured by Arduino and saves it to the MySQL database, also transmitting the data to Matlab clients connected through TCP/IP sockets.

The web interface has been coded in HTML with CSS and PHP server side scripts. On the client side, JavaScript is used to communicate with the server asynchronously to refresh data that is presented to the user and to send user commands to the server. When a data request is made, JavaScript on the client side transmits the command to an Apache web server, where PHP script receives that data and forwards it to the C++ code using sockets. The C++ code executes the command locally (for example, power down), or forwards it to Arduino, and then responds to the PHP script, which produces a response for the client. If the client requests measurement data, the PHP script reads the data directly from the MySQL server and sends formatted data to the client.

## References

1. Escalona, J.L., Recuero, A.M.: A bicycle model for education in multibody dynamics and real-time interactive simulation. Multibody Syst. Dyn. **27**(3), 383–402 (2012)
2. Whipple, F.J.W.: The stability of the motion of a bicycle. Quart J. Pure Appl. Math. **30**, 312–348 (1899)
3. Meijaard, J.P., Papadopoulos, J.M., Ruina, A., Schwab, A.L.: Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review. Proc. R. Soc. A **463**, 1955–1982 (2007)
4. Kooijman, J.D.G., Meijaard, J.P., Papadopoulos, J.M., Ruina, A., Schwab, A.L.: A bicycle can be self-stable without gyroscopic or caster effects. Science **332**(6027), 339–342 (2011)
5. Kooijman, J.D.G., Schwab, A.L., Meijaard, J.P.: Experimental validation of a model of an uncontrolled bicycle. Multibody Syst. Dyn. **19**(1–2), 115–132 (2008)
6. Frączek, J., Wojtyra, M.: Teaching multibody dynamics at Warsaw University of Technology. Multibody Syst. Dyn. **13**, 353–361 (2005)
7. Cavacece, M., Pennestrì, E., Sinatra, R.: Experiences in teaching multibody dynamics. Multibody Syst. Dyn. **13**, 363–369 (2005)
8. Pennestrì, E., Vita, L: Multibody dynamics in advanced educations. In: Ambrósio, J.A.C. (ed.) Advances in Computational Multibody Systems, pp. 345–370. Springer, Dordrecht (2005)
9. Fissete, P., Samin, J.C.: Teaching multibody dynamics from modeling to animation. Multibody Syst. Dyn. **13**, 339–351 (2005)
10. Fissete, P., Samin, J.C.: Engineering education in multibody dynamics. In: Bottasso, C.L. (ed.) Multibody Dynamics. Computational Methods and Applications, pp. 159–178. Springer, Dordrecht (2007)
11. Docquier, N., Fissete, P., Samin, J.C.: Hypotheses formulation in multibody dynamics: an educational challenge. Application to mountain bike dynamics. In: Multibody Dynamics 2009, ECCOMAS Thematic Conference, 29 June–2 July, Warsaw, Poland (2009)
12. de Jalón, G., Callejo, A.J.: Incorporating multibody dynamics into undergraduate subjects. In: 5th Asian Conference on Multibody Dynamics, 23–26 August, Kyoto, Japan (2010)
13. García de Jalón, J., Callejo, A.: A straight methodology to include multibody dynamics in graduate and undergraduate subjects. Mech. Mach. Theory **46**(2), 168–182 (2011)
14. Callejo, A., García de Jalón, J.: Teaching undergraduate numerical methods through a practical multibody dynamics project. In: Proceedings of the ASME Design Engineering Technical Conference, 4(Parts A and B), pp. 657–665 (2011)
15. Petuya, V., Altuzarra, O., Pinto, C., Hernandez, A: An educational software for the kinematic analysis of spatial mechanisms. In: Multibody Dynamics 2009, ECCOMAS Thematic Conference, 29 June–2 July, Warsaw, Poland (2009)
16. Rideout, G.: Teaching multibody dynamics in an undergraduate curriculum – an intuitive and explicit formalism based on parasitic elements. In: ASEE Annual Conference and Exposition, Conference Proceedings (2008)
17. Wolfsteiner, P.: Teaching multibody system simulation, an approach with Matlab. In: ASEE Annual Conference and Exposition, Conference Proceedings (2012)
18. Schwab, A.L.: On the interpretation of the Lagrange multipliers in the constraint formulation of contact problems: or why some multipliers are always zero? In: Proceedings of the ASME 2014 International Design Engineering Technical Conferences, DETC2014-34709 (2014)
19. Madgwick, S.O.H., Harrison, A.J.L., Vaidyanathan, A.: Estimation of IMU and MARG orientation using a gradient descent algorithm. In: Proceedings of the IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, 29 June–1 July 2011, pp. 1–7 (2011)
20. Valenti, R.G., Dryanovski, I., Xiao, J.: Keeping a good attitude: a quaternion-based orientation filter for IMUs and MARGs. Sensors **15**, 19302–19330 (2015)