Applications of Energy-Driven Computing: A Transiently-Powered Wireless Cycle Computer

Uvis Senkans, Domenico Balsamo, Theodoros D. Verykios, Geoff V. Merrett
Department of Electronics and Computer Science
University of Southampton, UK
{us1n16,d.balsamo,t.verykios,gvm}@ecs.soton.ac.uk

ABSTRACT

There has been a dramatic increase in recent years in the number of battery-powered embedded electronic devices. However, the lifetime of these devices is limited by battery capacity. Energy harvesting is an efficient solution to overcome this limitation; however, large energy buffers have been traditionally employed to tackle the temporal variation of the source. These buffers typically require considerable time to charge while introducing a cost, size and weight overhead. Energy-driven systems are specifically designed to operate from an energy harvesting source, without overprovisioning energy storage to make the system appear "battery-like". Furthermore, a transiently-powered system is capable of sustaining computation despite an intermittent supply, without the need for additional energy storage. While this shows much promise, the wide applicability of these systems to real-life applications is yet to be demonstrated. This paper presents a transiently-powered wireless bicycle trip counter which measures distance, speed and active cycling time, and transmits data wirelessly. The system sustains operation by harvesting energy from the rotation of the wheel, operating from a minimum speed of 6kph, while adapting its operation in response to the harvested energy.

CCS CONCEPTS

• **Hardware → Sensors and actuators**; Wireless devices;

KEYWORDS

Transient Computing; Embedded Systems; Energy Harvesting

1 INTRODUCTION

Batteries have traditionally been used to power embedded electronic devices. However, modern embedded systems typically require a long lifetime, low cost, and low weight, which poses significant issues to battery-powered systems. In addition, a principal challenge for wearable and autonomous systems is the maximisation of time they can operate without human intervention such as battery recharging or replacement [4]. For this reason, the powering of electronic devices from harvested energy has gained increasing interest in recent years [5].

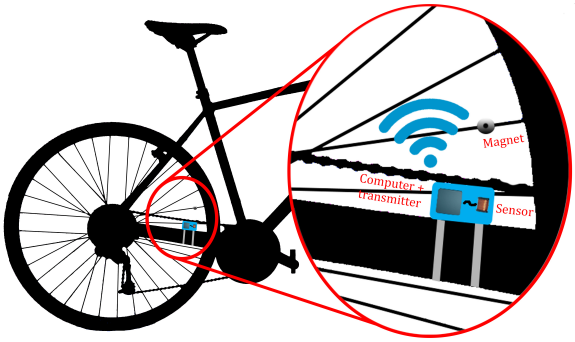

Figure 1: Concept diagram of transiently-powered wireless cycle computer.

Energy harvesting systems scavenge energy from environmental sources such as light, vibration, motion or temperature to power themselves, instead of relying on batteries [5]. However, the temporal and spatial conditions can have a considerable effect on the stability of the source. Relying solely on the harvested energy can, therefore, result in the system being unable to sustain computation. Energy buffers such as rechargeable batteries or supercapacitors are usually introduced in order to cope with the variability of the source. However, the size, weight and cost of the system is significantly increased with the addition of the buffer.

Energy-driven systems require the energy storage added to the system to be minimised. This typically requires computation to be sustained despite the variable and intermittent energy (referred to as a "transient" source) harvested from the environment [1]. These systems are designed in such a way that the energy availability is taken into equal consideration with the application's requirements.

The WISPCam device [6] is a wireless camera which uses scavenged RF energy as energy source. However, it requires the addition of a large energy storage (e.g. a 6mF supercapacitor) which enables the system to take single photo and store it in NVM.

A self-powered piezoelectric energy harvester was presented by Vasic et al. [8]. This paper focused on potential applications of self-powered systems which scavenge energy from the rotation of a wheel, such as powering a small on-board computer or a low-energy flash light. However, the characterisation of the source was the main contribution this work, without applying the results to realise a real-life application.

A self-powered wireless speed sensor was proposed by Buccolini et al. [2] which uses an induction coil and a magnet to harvest

ENSsys'17, November 5, 2017, Delft, Netherlands

ACM ISBN 978-1-4503-5477-6/17/11…$15.00
https://doi.org/10.1145/3142992.3142993

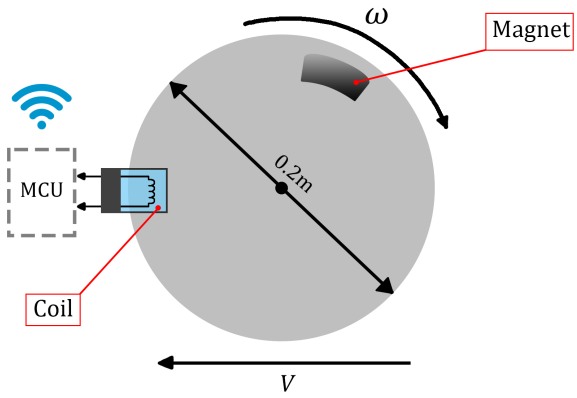

Figure 2: Diagram of experimental setup

energy from the rotation of the wheel. However, this system does not consider changes in energy availability and its operation requires the system to be powered up continuously. Therefore, it is unable to operate in a transient scenario and its design involves a relatively large capacitance ($1500\mu F$) to ensure the continuous supply of energy.

However, few real-life applications have been implemented using energy-driven systems to date, due to their design complexity and the limited energy budget that they normally offer. Examples of such systems are a transiently-powered energy harvesting step counter for integrated wearable applications [7] and an energy harvesting approach to energy metering, where the energy harvesting power supply acts as the sensor [3].

In this paper, we report on a novel application of energy-driven computing: a transiently-powered wireless cycle computer counter which operates directly from a kinetic energy harvester, with a minimum of energy storage. The system operates on every wheel rotation, adjusting its mode of operation depending on the available energy. The remainder of this paper is organised as follows: Section 2 describes the design of the system, including the characterisation of the harvester. The experimental validation of the system is presented in Section 3 and the paper is concluded in Section 4.

2 ENERGY-DRIVEN CYCLE COMPUTER

The energy harvester of the wireless energy-driven cycle computer consists of a magnet attached on the wheel and a coil which is affixed to the frame of the bicycle. Compared to traditional systems, the coil in this case works both as the energy source and sensor, which helps in the miniaturisation and the cost reduction of the system. Figure 1 shows the concept of our system. Energy is harvested from the rotation of the wheel and the system is able to adapt its operation depending on the energy availability.

To design this system, we need to analyse the design principles and constraints for the proposed energy-driven cycle computer. The performance of the kinetic energy harvester is considered (Section 2.1), followed by the hardware design of the system (Section 2.2). Finally, the different operational modes of the system depending on the harvested energy are discussed (Section 2.3).

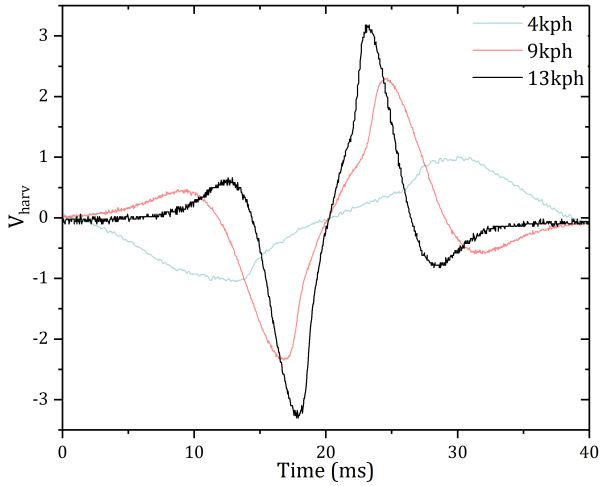

Figure 3: Output voltage (V_{harv}) at different speeds

2.1 Kinetic energy harvester

The performance of the kinetic energy harvester is discussed in this section, to characterise the voltage and power output and establish the relationship between them and travelling speed. This aids the design of the wireless cycle computer in section 2.2.

The system replaces the reed switch (used to detect a wheel rotation) of a traditional cycle computer with a coil which is used to detect rotation and harvest energy. In this case, an electromagnetic field is created at the coil using a magnet (which is rotating with the wheel), inducing an electromagnetic force across the coil (V_{coil}). This voltage is used to estimate the speed, distance and active cycling time of the bicycle, while the generated energy powers the system. The amplitude of the pulse and the frequency between pulses are directly proportional to the speed. Faraday's law describes the voltage across the coil (equation 1):

$$V_{coil} = -\frac{\partial \Phi_B}{\partial t} \quad (1)$$

The rate of change of the magnetic flux depends on the speed of the wheel. Therefore, the peak voltage is increased with increasing speed, while the duration of the pulse decreases.

An experimental setup was created to enable the characterisation of the energy harvester, as shown in Figure 2. Here, a disc with a diameter of 0.2 metres rotates and a magnet attached to the disc induces an electromagnetic force across the coil. We measure this force at different speeds to characterise the harvester.

The typical output voltage of the harvester (V_{harv}) at three different speeds (4, 9 and 13 kph) is shown in Figure 3. Here, we demonstrate the effect of Faraday's law on the harvester of the system, as the period of the voltage pulse decreases with increasing speed, while its amplitude increases.

Figure 4 shows the output power of the energy harvester at three different speeds (4, 9 and 13 kph), with a load of 1kΩ which is typical for a modern microcontroller. The highlighted area under each curve corresponds to the generated energy after the magnet crosses the coil. In this case, the output energy increased more than

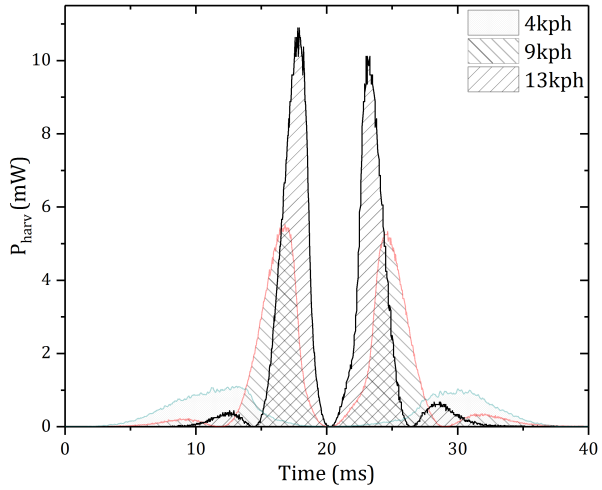

Figure 4: Output power (P_{harv}) at different speeds

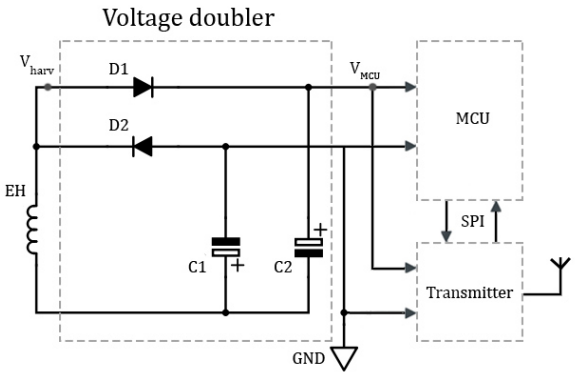

Figure 5: Schematic of the proposed system.

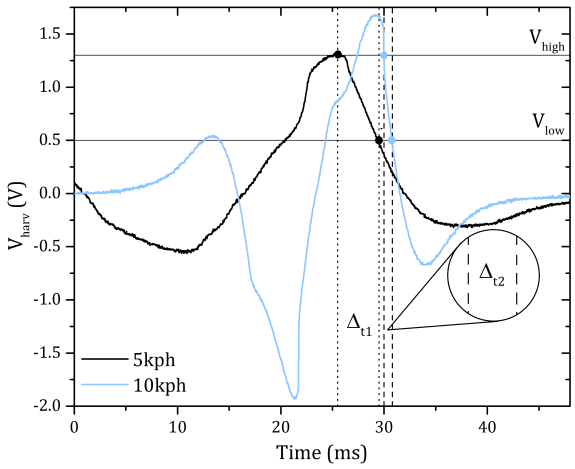

Figure 6: Calculation of speed using V_{harv}.

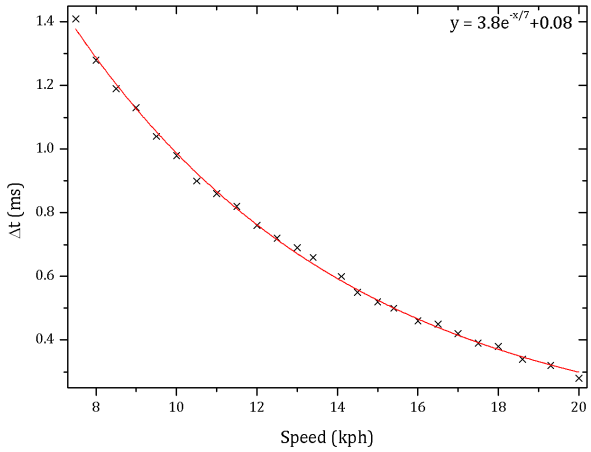

Figure 7: Correlation between speed and Δt, showing line of best fit.

7x between 4kph and 13kph (6.6μJ and 48.2μJ) which enables us to execute different tasks depending on the available energy.

2.2 Hardware Design

With the aid of the characterisation of the harvester presented in Section 2.1, we are able to design the wireless energy-driven cycle computer, as shown in Figure 5.

A rectifier converts the AC output from the harvester to DC in order to power the components of the system. According to Figure 3, the peak voltage at lower speeds (<2V) is not sufficient to power a typical microcontroller (MCU). For this reason, a simple voltage doubler is used to boost the input voltage without significantly increasing the design complexity and area. An exploration into the effect of the size of C1 and C2 is given in Section 3.3.

An MSP430FR5739 microcontroller (MCU) calculates the speed of the bicycle. As mentioned in section 2.1, the duration of the output voltage pulse of the harvester (V_{harv}) is affected by the speed of the bicycle. Therefore, we obtain the speed by measuring the time between two predefined voltage points (V_{high} and V_{low}) on the decay of harvester pulse using an ADC window, as shown by Δ_{t1} and Δ_{t2} in Figure 6. The system does not have a common ground, meaning that V_{harv} and V_{mcu} do not share the same ground. Therefore, V_{mcu} has a voltage offset of approximately 0.7V which is caused by the Schottky diodes used in voltage doubler circuit.

Figure 7 illustrates the correlation between travelling speed and Δt. As speed is increasing, Δt is decreasing because the period of V_{harv} is becoming smaller, similar to Figure 3. The decrease of Δt is exponential as the magnetic flux induction rate across the coil is non-linear for different speeds. As the speed calculation depends on Δt, data must be processed according to the line of best fit, as shown in figure 7, in order to obtain accurate measurements.

The distance covered by the bicycle is measured by increasing a variable at every generated pulse with the value of the circumference of the wheel. The total distance is saved to NVM before a power intermission, and is restored once power is available again so that it can be updated.

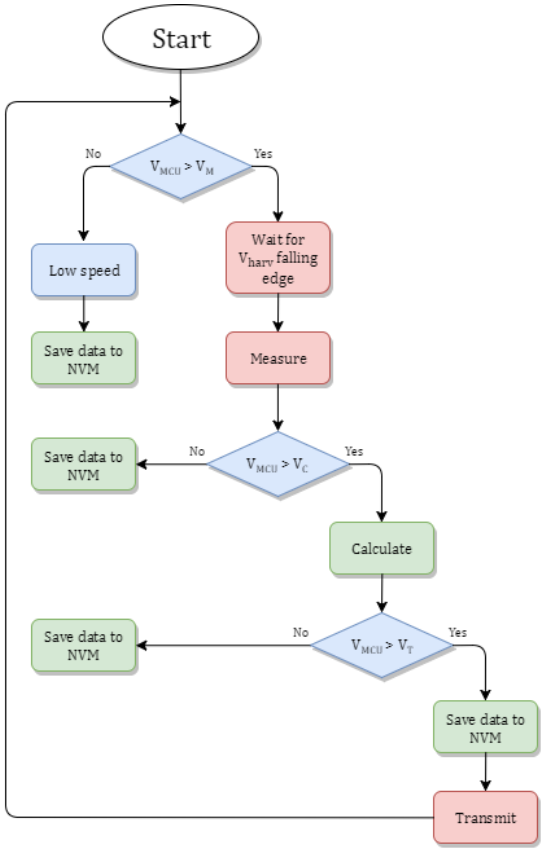

Figure 8: Flowchart of the system.

Due to the transient nature of the source, the system has no sense of time. Therefore, the active cycling time calculation is based on the relationship between speed, distance and time. Given the circumference of the wheel, it is possible to estimate the time it takes for a single revolution of the wheel when travelling at a certain speed. The total active time is saved to NVM and it gets updated on every revolution, similar to the distance counter.

These data are transmitted wirelessly to a nearby device (e.g. smartphone) using an ultra low power transceiver (nRF24L01). To communicate with the nRF24L01 transceiver module, a 4-wire SPI is used. When valid data is processed by the MCU, it passes them to the transmitter which handles the data packet assembly using the ShockBurst communication protocol. Each packet consists of a preamble byte for data validity, 5 address bytes and 1-32 payload bytes following by CRC byte. To receive data, a battery powered nRF24L01 transceiver is configured in receiver mode and is constantly listening using the same protocol. Whenever a payload is received, it is sent to serial terminal via UART.

Table 1 shows the average duration and energy consumption for each task that the system needs to execute. If there is not enough energy for a task to be completed, the system saves the necessary data to NVM so that it can be restored once the available energy is sufficient, as explained in Section 2.3.

Task	Speed (kph)	Duration (ms)	Energy (μJ)
Wait (NOP)	8.3	0.63	0.30
	12	4.96	2.62
	16	3.52	2.18
Measure	8.3	1.31	4.12
	12	0.78	2.68
	16	0.50	2.02
Calculate	8	0.35	1.09
	12	0.35	1.19
	16	0.35	1.40
Transmit	8.3	0.90	16.1
	12	0.90	17.6
	16	0.90	20.7

Table 1: Task-specific time and energy requirements.

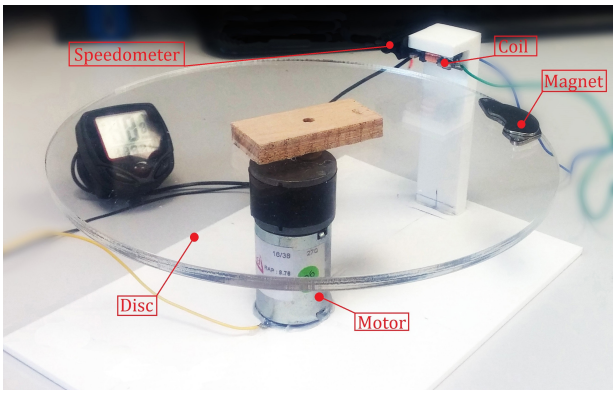

Figure 9: Experimental test setup of the system.

2.3 Operational Modes

As explained in section 2.1, the available energy is tightly coupled to the speed. As this is an energy-driven system, the system applies different operational modes at different speeds to manage and scale performance.

While travelling at very low speeds the scavenged energy is not enough to complete all required tasks in one iteration. To accumulate available energy at low speeds and save valid data, the system is designed to operate in 4 different modes - low speed, measure, calculate and transmit. There are 3 different preset threshold voltages characterising the boundaries of these modes - V_M, V_C, and V_T. These voltages are compared to V_{mcu} using an ADC, and depending on the outcome, the system proceeds with the execution of the task or saves the necessary data to NVM.

Figure 8 shows a flowchart of the system. Low power mode is active when the travelling speed is less than the minimum speed for speed measurement, i.e. the voltage of the MCU is less than threshold V_M. In this case, the system records the speed as "low" and saves this to NVM. Measure mode is used to measure the time (Δt)

$V_{threshold}$ (V)	V_{MCU} (V)	Speed (kph)	Mode
V_{LS}	2.02	>6.4	Low speed
V_M	2.09	>7.3	Measure
V_C	1.96	>7.5	Calculate
V_T	1.96	>7.9	Transmit

Table 2: Threshold voltages and minimum speeds for 50μF.

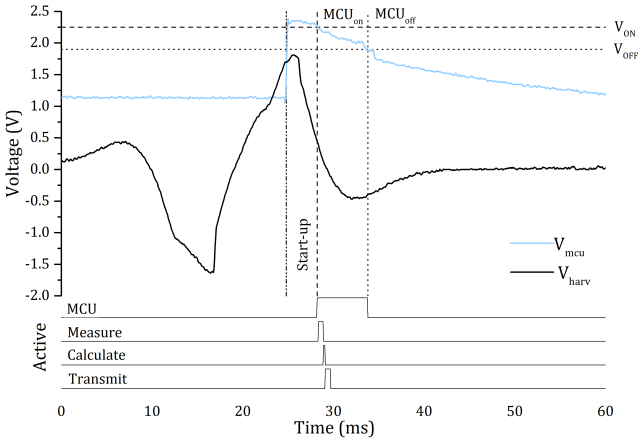

Figure 10: Operation of the the system at 8kph.

between V_{high} and V_{low} at the falling edge of the generated pulse (V_{harv}), once V_{MCU} exceeds V_M. Unless there is enough energy to enter the following mode (V_{MCU}>V_C), the system saves the measured time to NVM, otherwise it enters Calculation mode. In Calculation mode, the system uses the data obtained from Measure mode to calculate speed, distance and active cycling time. The energy availability is checked again (V_{MCU}>V_T) to decide whether the calculated will be transmitted wirelessly (Transmit mode) or saved to NVM.

3 EXPERIMENTAL VALIDATION

An experimental test setup was created, as shown in Figure 9. Here, a magnet is attached to a rotating disc (representing the wheel) controlled by a motor. A speedometer (SB-318) measures the actual speed for comparison with experimental results. Four GPIOs are configured to indicate the different stages of operation (MCU off, measure, calculate, transmit) for debugging.

3.1 Thresholds

Table 2 shows the threshold voltages (V_{LS}, V_M, V_C and V_T) and minimum speeds required for each mode for a system with 50μF capacitance. The system starts operating at a speed greater than 6.4kph, while it is able to wirelessly transmit data at 7.9kph.

3.2 System Operation and Accuracy

The operation of the proposed system at a constant speed of 8kph is shown in Figure 10. Once the system detects a voltage greater than V_{th} capable to run the microcontroller, the MCU is configured and

Speed (kph)			Distance (m)			Active Time (s)		
A*	M**	Error	A*	M**	Error	A*	M**	Error
8.0	7.4	8.1%	101.4	98.3	3.1%	45.6	48.2	5.4%
10.1	10.0	1.0%	127.7	126.7	0.8%	45.3	45.0	0.7%
12.0	12.0	0.0%	151.4	151.9	0.3%	45.2	45.2	0.0%
14.1	14.1	0.0%	177.2	176.4	0.4%	45.2	44.5	1.6%

* A = Actual values **M = Measured values

Table 3: Accuracy measurements at different speeds.

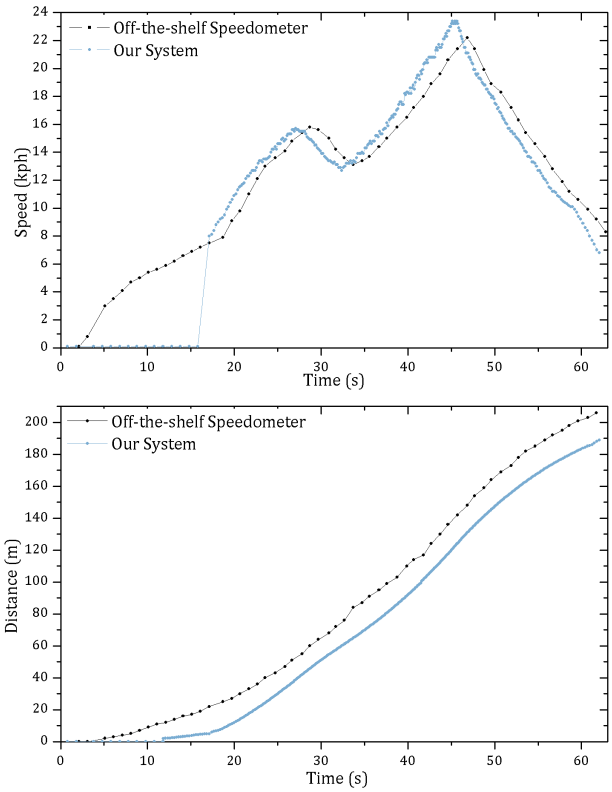

Figure 11: Speed (a) and distance (b) profile over time comparison.

the sensor starts collecting information. Subsequently, the MCU calculates the speed, distance and active cycling time which are transmitted to the receiver. The system can sense and save data to NVM at 6kph, while a speed of 8kph is required for wireless transmission.

Table 3 shows results on the measurement accuracy of the proposed system at various speeds. Here, we observe that the error in measurements is higher at lower speeds, while the accuracy increases with increasing speed. This is due to the fact that the voltage induced across the coil is not of linear nature. However, as the travelling speed increases, the system samples data at a higher rate which results in offering better granularity.

Figure 11 shows the speed (a) and distance (b) profile of the wheel over time, as captured by the speedometer and the transiently-powered wireless cycle computer. The speedometer has a persistent

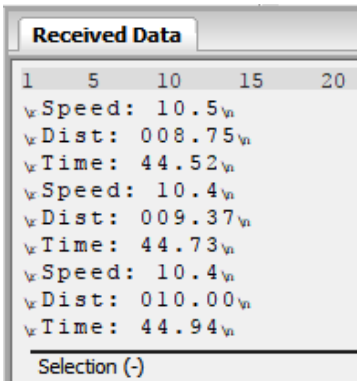

Figure 12: Screenshot of received data at serial terminal.

delay of 1.3 seconds compared to our system. The speedometer's sampling rate is 1Hz, whereas our system's sampling rate ranges between 3.5 and 10.6Hz, depending on travelling speed. Consequently, the presented system offers improved granularity and can detect more rapid changes in speed compared to the traditional reel switch speedometer due to its ability to record changes on every rotation of the wheel. However, the proposed system is unable to measure the exact speed until there is enough energy for the system to enter "measure" mode.

For the distance measurement, the delay of the speedometer has decreased as its sampling rate is higher and, therefore, it lags for less than a second compared to the wireless energy-driven cycle computer. However, our system starts measuring the distance travelled after the cycle reaches 6.4kph (low speed mode) and therefore, it is unable to measure the metres travelled below this speed.

An example of communication between the transmitter and the receiver is shown in Figure 12. Here, we show the data received at the serial terminal using UART, showing the speed, distance and active cycling time for three consecutive wheel rotations.

Our experimental setup uses a wheel of 0.2m in diameter, while typical bicycle wheel diameters range from 0.5 to 0.65m. For the same linear speed, angular velocity changes according to the diameter. Therefore, the speed calculation would remain unaffected, while the distance calculation would need to reflect the change in diameter. Finally, the frequency of sampling (generated voltage pulses) would be decreased, resulting in lower granularity, without affecting the accuracy of measurements.

3.3 Capacitance

Greater capacitance acts as an energy reservoir, enabling system to enter its operating modes at lower travelling speeds. Figure 13 shows the minimum required speed for each operating mode, for different values of added energy storage. A capacitance of $50\mu F$ allows entering all modes at a speed lower than 8kph. However, a capacitance of $235\mu F$ has a negative effect on the system as the MCU boots up only when travelling at speeds greater than 8 kph. Due to the large capacitance, V_{mcu} increases at a lower rate and therefore, it requires a higher speed to reach its threshold. Furthermore, in this case, the MCU is powered on later and consequently, the measurement cannot be completed as the threshold voltages (V_{high} and V_{low}) are missed.

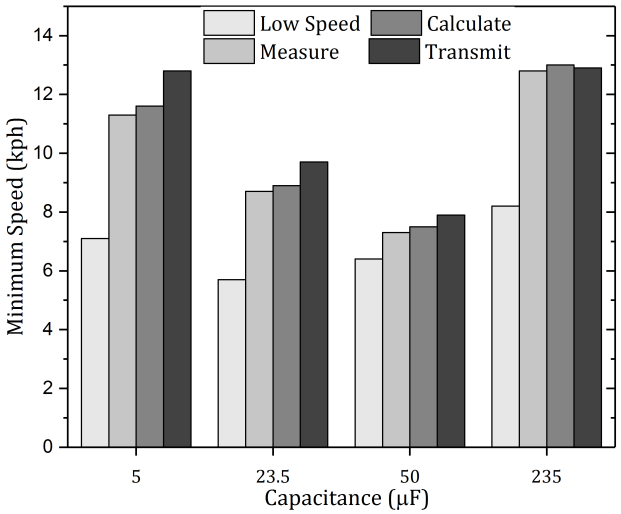

Figure 13: Minimum speed of operation for different capacitances.

4 CONCLUSION

In this paper, a novel transiently-powered wireless cycle computer has been presented as a demonstrator of a real-life energy-driven system. The system operates directly from the energy harvested from the rotation of the wheel with a minimum of added energy storage. The system has been experimentally validated, showing successful operation from an intermittent supply at different speeds. Our future work includes improving the networking part of the system and fitting the system on a real bicycle.

ACKNOWLEDGMENTS

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) under EP/P010164/1. Experimental data used in this paper can be found at DOI:10.5258/SOTON/D0260 (http://doi.org/10.5258/SOTON/D0260).

REFERENCES

[1] D. Balsamo, A. S. Weddell, A. Das, A. R. Arreola, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini. 2016. Hibernus++: a self-calibrating and adaptive system for transiently-powered embedded devices. *IEEE Trans. Comp.-Aided Des. Integr. Circuits Syst.* 35, 12 (2016), 1968–1980.

[2] L. Buccolini and M. Conti. 2017. An Energy Harvester Interface for Self-Powered Wireless Speed Sensor. *IEEE Sensors Journal* 17, 4 (2017), 1097–1104.

[3] S. DeBruin, B. Campbell, and P. Dutta. 2013. Monjolo: An energy-harvesting energy meter architecture. In *11th ACM Conference on Embedded Networked Sensor Systems 2013*. ACM, Rome, Italy, 18.

[4] D. Ferreira, A. Dey, and V. Kostakos. 2011. Understanding human-smartphone concerns: a study of battery life. *Pervasive Computing* (2011), 19–33.

[5] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green. 2008. Energy harvesting from human and machine motion for wireless electronic devices. *Proc. IEEE* 96, 9 (2008), 1457–1486.

[6] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith. 2015. Wispcam: A battery-free rfid camera. In *RFID (RFID), 2015 IEEE International Conference on*. IEEE, 166–173.

[7] A. Rodriguez, D. Balsamo, Z. Luo, S. Beeby, G. Merrett, and A. Weddell. 2017. Intermittently-powered energy harvesting step counter for fitness tracking. In *IEEE Sensors Applications Symposium (SAS) 2017*. IEEE, Glassboro, NJ, USA, 1–6.

[8] D. Vasic, Y. Y. Chen, and F. Costa. 2014. Self-powered piezoelectric energy harvester for bicycle. *Journal of Mechanical Science and Technology* 28, 7 (2014), 2501–2510.