Practical Control of Transmission Power for Wireless Sensor Networks

Yong Fu*, Mo Sha*, Gregory Hackmann, Chenyang Lu
Department of Computer Science and Engineering, Washington University in St. Louis, USA
{fuy, sham, gwh2, lu}@cse.wustl.edu

***Abstract*—Transmission power control (TPC) has the potential to reduce power consumption in Wireless Sensor Networks (WSNs). However, despite a multitude of existing protocols, they still face significant challenges in real-world deployments. A practical TPC protocol must be *robust* against complex and dynamic wireless properties, and *efficient* for resource-constrained sensors. This paper presents P-TPC, a practical TPC protocol designed on control-theoretic techniques. P-TPC features a highly efficient controller designed on a dynamic model that combines a theoretical link model with online parameter estimation. P-TPC's robustness and energy savings are demonstrated through trace-driven simulations and real-world experiments in a campus building and residential environments.**

I. Introduction

Transmission power control (TPC) has received attention as a promising approach to save power in Wireless Sensor Networks (WSNs). However, despite a multitude of existing protocols, TPC faces significant challenges in real-world deployments, where wireless links exhibit complex and dynamic characteristics. To be practical, a TPC protocol must be *robust*, able to maintain satisfactory link quality under varying conditions. It must also be *efficient*, incurring minimum processing, communication, and memory overhead on embedded sensors. Given the complex dynamics of wireless links, a TPC protocol needs to be designed and analyzed carefully to quickly adapt to changing conditions without causing instability in link quality.

This paper presents a practical TPC protocol, *P-TPC*, that meets the above challenges by combining a practical system design and a rigorous control-theoretic approach. Specifically, the contributions of this work are four-fold.

- A highly efficient *feedback control algorithm* that maintains a desired link quality by dynamically adjusting transmission power based on online link quality measurements;
- *Stability analysis* that ensures convergence to link quality set points under a range of wireless conditions;
- An *online link model identification* scheme that captures changes to link dynamics and triggers controller reconfiguration when needed;
- A light-weight implementation and empirical evaluation of P-TPC on the TinyOS and TelosB mote platforms;

The robustness and energy savings under P-TPC have been demonstrated by both simulations based on wireless traces collected from a campus building, as well as experiments on a physical wireless sensor network testbed.

The rest of the paper is organized as follows. Section II reviews related work. Section III provides an overview of the P-TPC protocol. As a foundation for the control design, Section IV presents the dynamic link model and the online model identification scheme used to estimate the model parameters at run time. Section V provides the control design and analysis. Section VI provides an evaluation based on both real-world traces and experiments on real networks in a campus building and residensial environments. Section VII concludes the paper.

II. Related Work

A multitude of TPC protocols have been proposed to maintain certain measure of link quality by adjusting transmission power dynamically. Existing protocols can be classified along two dimensions: (1) the metric used to control link quality and (2) the algorithmic design approach.

A direct metric of the quality of a link is *packet reception ratio* (PRR), i.e., the fraction of transmitted packets that have been successfully received by the receiver. However, most existing TPC protocols employ other metrics as proxies of PRR in their control design to avoid the overhead of measuring PRR at run time. ATPC [18] uses Received Signal Strength Indication (RSSI) or Link Quality Indicator (LQI) as a proxy for PRR. Using profiling data acquired offline, a lower bound on PRR is converted into a lower bound on RSSI or LQI. At runtime a feedback loop is used to estimate a linear relationship between the transmission power and the RSSI/LQI at the receiver. Based on the estimated linear relationship, ATPC selects the proper transmission power to achieve the necessary lower bound on RSSI/LQI and, indirectly, the lower bound on PRR. However, empirical studies found that RSSI and LQI are not always robust indicators of PRR [14]. As an extension of ATPC, in [19], the authors proposed a hierarchical framework to improve throughput and energy efficiency, where transmission power control is a part of link layer protocol to maintain link quality and optimize performance. This approach shares the same shortcoming with ATPC in that it relies on indirect metrics that are unreliable proxies for PRR.

Control-theoretic approaches have been proposed to deal with TPC in recent literature including robust control [4], [27], [28], model predictive control [29] and Markov chain [5]. These control-theoretic approaches are based on RSSI measurements and thus suffer from the same robustness issues as

*The first two authors made equal contribution to this work. This work is supported in part by NSF through grants CNS-1035773 (CPS), CNS-1144552 (NeTS) and CNS-0448554 (CAREER).

	Control-Theoretic	Heuristic
Direct Link Quality Metric	P-TPC	[14], [23]
Indirect Link Quality Metric	[4], [5], [18], [19], [27]–[29]	[6], [7], [9], [12], [16], [17]

TABLE I
TAXONOMY OF TRANSMISSION POWER CONTROL.

ATPC. Moreover, these control algorithms are computationally intensive for resource-constrained embedded sensor platforms.

In contrast to previous works based on indirect link quality metrics, PCBL and ART works with direct measurements of PRR. PCBL [23] relies on extensive profiling of PRR at all power levels and then select the transmission power for links in the network. PCBL further assumes high-quality links also tend to be highly stable, which was found not always hold in complex environments [14]. As a result, PCBL is not suitable for dynamic environments where link properties change significantly over time. ART [14] is designed to adjust transmission power based directly on PRR observed at runtime. ART computes each link's PRR over a sliding window and compares it against an application-specified target. However, ART employs a simple feedback-based heuristic to adjust transmission power that results in overly conservative behavior compared to our approach (as shown in Section VI).

We summarize the above related works in Table I according to design approach and link quality metrics. To our best knowledge, P-TPC is the first protocol that applies a control-theoretic approach to control PRR directly. Since P-TPC does not assume a fixed relationship between PRR and RSSI/LQI, it can deliver reliable performance in environments where RSSI and LQI are poor proxies for PRR. At the same time, P-TPC employs a control-theoretic design that both robust and efficient suitable for dynamic wireless environments and resource-constrained platforms. To further enhance the robustness P-TPC can be reconfigured on-the-fly to accommodate the significant dynamics of real-world wireless link behavior.

Early works on topology control focused on manipulating the geometric properties of WSNs through transmission power control [6], [7], [9], [12], [16], [17], [22]. They usually rely on ideal wireless models with unrealistic assumptions such as circular transmission range which do not apply to realistic indoor environments [10]. In contrast, P-TPC employs a practical system approach to directly control link quality based on online measurements. The robustness of P-TPC has been demonstrated via implementation on real hardware and experiments on a physical testbed.

III. PROTOCOL OVERVIEW

P-TPC is a receiver-oriented protocol: i.e., receivers select the transmission power for all incoming links. As shown in Figure 1, for each incoming link P-TPC runs a feedback control loop in the end of each sampling period. It monitors all incoming packet transmissions and keeps a record of the total number of transmissions and failures within the current sampling period. In the end of a sampling period, P-TPC calculates the PRR in the sampling period $y(k)$ (where k represents the kth sampling period), computes the transmission power $p(k+1)$ that should be used by the sender in the next sampling period, and sends $p(k+1)$ to the sender.

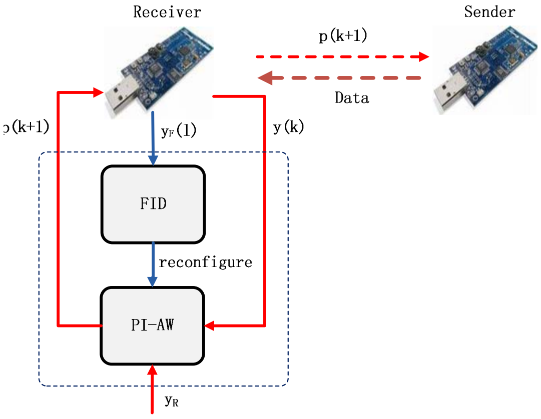


Fig. 1. P-TPC Structure.

P-TPC consists of two major components: a *Fast on-line model IDentification* (FID) and a *Proportional-Integral with the Anti-Windup* (PI-AW) controller. FID efficiently estimates the model between PRR and transmission power and then initializes or reconfigures PI-AW. PI-AW computes the transmission power based on the difference between the current PRR and an application-specified PRR set point. It adopts a simple but robust *Proportional-Integral* (PI) control algorithm, and an *Anti-windup* (AW) component which is responsible for the nonlinear part of the transmission power model. PI-AW adapts its parameters on-line in accordance with FID outputs.

A *bootstrapping* phase is employed to initialize the power model parameters before starting the feedback loop responsible for manipulating transmission power. In the bootstrapping phase, each link is initially set to transmit at its maximum power, and then transmits a sequence of probe packets from highest to lowest power level to profile the transmission power model. Once link quality data is collected, FID uses the data to estimate the link model and PI-AW is configured from this initial model.

There are two feedback loops in P-TPC. The inner loop involving the PI-AW controller is responsible for adapting transmission power based on PRR measurements. The outer loop involving FID is responsible for adjusting the parameters based on the updated power model. To reduce overhead and stabilize the feedback loop, the sampling periods of the outer loop , T_m, should be larger than the sampling period of the inner loop, T_p. The choice of the sampling periods need to consider the tradeoff between run-time overhead and the responsiveness to changes of link quality and power model. The sampling periods of the inner and outer loops are set to 5 minutes and 150 minutes, respectively, in our implementation.

At the beginning of the FID feedback loop's lth sampling period, probe packets are sent from the sender node and the PRR $y_F(l)$ is collected. The FID is then activated to identify the parameters of the power model in this sampling period. The PI-AW controller updates its parameters based on the identified model and uses therm to compute transmission power control in the inner loop. At the end of the kth sampling period of the inner loop, the PI-AW controller is invoked and computes the required power level of the sender node based on the PRR $y(k)$ measured by the receiver node and PRR set point y_R. The receiver sends it to the sender node, which in turn set its transmission power accordingly.

Note that the outer loop is designed to handle significant variations in the relationship between transmission power and PRR (i.e., mode changes), whereas the inner loop is designed to directly adapt to changes of PRR over a shorter term. When the power model does not change significantly, the outer loop can be executed only once to bootstrap and configure the PI-AW controller in the beginning. This self-tuning feature can be useful in practice as users are usually not control engineers.

IV. Model and Identification

As a foundation for our control design, we present a dynamic model that characterize the relationship between the transmission power and the PRR, followed by the description of an efficient system identification algorithm used by FID to estimate the model parameters at run time.

A. Link Model

The link quality of a link from the sender n_i to the receiver n_c is represented by its PRR, denoted by $y(k)$ in the k^{th} sampling period. A sender n_i may increase its transmission power level to improve the PRR, or decrease the transmission power to save energy when the link quality is good.

Corresponding to the behavior of common radio hardware [3], we assume that senders may adjust the transmission power to one of m discrete power levels: i.e., the available power levels are defined over a discrete set $\mathcal{P} := \{p_k | 1 \leq k \leq m, p_k \in N\}$. Previous work has suggested that a piecewise linear relationship can be used to approximate the actual power output from a specified power level [4], [8].

Although in this work we follow similar steps to build the link model as in [20], [30], our model focuses on the dynamic behavior of PRR, while [30] aims to analyze the transitional relationship between link distance as well as link quality and [20] studies PER (Packet Error Rate).

Radio propagation is commonly modeled using the log-normal shadowing path model

$$PL(d) = PL(d_0) + 10n\log_{10}(\frac{d}{d_0}) + X_\sigma, \tag{1}$$

where $PL(d)$ is the path loss at distance d, $PL(d_0)$ is the path loss at a reference distance d_0, n is the path loss exponent (signal decaying rate), and X_σ is a zero-mean Gaussian random variable with the standard deviation σ [26]. While the radio propagaiton model (1) may suffer inaccuracy in complex indoor environment, the feedback control approach employed in this work is robust to such model errors as shown in our experimental results in Section V.

At the k^{th} sampling period, the SNR $\gamma_i(k)$ at the receiver may be written as

$$\gamma_i(k) = p_i(k) - PL(d_i) - p_n \tag{2}$$

where $p_i(k)$ is the output power of the sender n_i, $PL(d_i)$ is the path loss defined by (1), and p_n is a constant noise floor [30].

The relationship between SNR and bit error rate depends on the modulation and encoding scheme used by the underlying radio platform and can be obtained from literature [30]. In our experiments, we use the Chipcon CC2420 radio, which employs the modulation and encoding specified by the IEEE 802.15.4 standard[1]. For this radio, the bit error rate P_e may be modeled as

$$P_{e,i}(k) = \frac{1}{2}\left(1 - \sqrt{\frac{\gamma_i(k)}{1+\gamma_i(k)}}\right) \tag{3}$$

Assuming bit errors are independent and randomly distributed, the PRR can be written as

$$y_i(k) = (1 - P_{e,i}(k))^l, \tag{4}$$

where l is the bit length of the packet. Combining (3) and (4), we thus derive the relationship between transmission power and PRR

$$y_i(k) = \left(1 - \frac{1}{2}\left(1 - \sqrt{\frac{\gamma_i(k)}{1+\gamma_i(k)}}\right)\right)^l. \tag{5}$$

Combining Equation (2) and Equation (5), we are able to acquire the link model between PRR and transmission power.

For this nonlinear model, it is difficult to design an efficient model identification and transmission power control algorithm. However, the analytical model derived above and in previous studies [25], [30] show the relationship between PRR and transmission power can be approximated by bimodal behavior associated with transmission power. Specifically, PRR is consistent when SNR exceeds a high threshold or falls below a low threshold, and is variable inside the *transitional region* between these two thresholds. To reduce the model complexity but still capture major behavior, we could approximate the model (5) by a simple piecewise linear model, which is aggregation of three linear models for each of the three regions, beyond the high threshold, below the low threshold and between the high and low thresholds, respectively. Formally, for the link between a sender n_i and the receiver n_c, the linear model between PRR and power can written as

$$y_i(k) = \begin{cases} y_{u,i}, & \text{if } p_i \leq p_{u,i}, \\ y_{l,i}, & \text{if } p_i \geq p_{l,i}, \\ g_i p_i(k), & \text{otherwise}, \end{cases} \tag{6}$$

where p_u, p_l are threshold power levels, g_i is the gain between the PRR and the power, and $y_{u,i}$ and $y_{l,i}$ are upper and lower bounds on PRR. $p_{u,i}, p_{l,i}$ can be seen as the *soft* constraints of

[1] Although our discussion is based on the CC2420, we emphasize that P-TPC could be adapted for other radio hardware by substituting the appropriate bit error rate model from literature.

the power levels, which confines the PRR in a reasonable area and beyond which energy is wasted since increasing power can not improve the link quality marginally. We will present a highly efficient model identification algorithm to identify the parameters $y_{u,i}$, $y_{l,i}$, and g_i in the following section.

B. Online Model Identification

An online model identification approach makes P-TPC more tolerant to inaccuracies of the analytical model (e.g., impacts of walls and noise on signal propagation). Moreover, it can be used to automatically reconfigure the transmission power controller, an approach adopted by P-TPC .

Common model identification methods identify model and parameters via optimization. However, for resource-constrained wireless sensors, these methods incurs impractical computational overhead. Hence, we design an efficient model identification algorithm, FID, by leveraging the characteristics of the piecewise linear model (6). Only 4 parameters (p_l, y_l, p_u, y_u) are needed for the link model (6), and they decide two *turning points* of the piecewise linear model. Based on this observation, the following rules are employed to identify the parameters:

Rule 1: If the maximum difference between PRRs of any two transmission power levels is lower than a threshold, e.g., 10%, then we consider that link quality is stable (may be good or poor) over the entire power level set. So in this case the model can be reasonably approximated as a single horizontal line. i.e., $p_u = p_{max}, p_l = p_{min}$ and $y_l = y_u$.

Rule 2: If the PRR of the lowest power level is higher than a threshold, for example 90%, then the model is a straight line since in this case the link quality is stable and good for all power levels.

Rule 3: For the other cases, we first derive the difference between PRR for all power levels. Then the power level and the PRR corresponding to the the minimum PRR difference is chosen as (p_l, y_l) and the power level and the PRR corresponding to the maximum one as (p_u, y_u).

Although we can add more rules to improve accuracy of the results of model identification, our experiments indicate only these three simple rules can achieve satisfactory results. To reduce the impact of noises in the measured PRR, before applying these rules, the data for model identification is preprocessed by a *moving average* function.

Compared to commonly-used model identification algorithms, such as *Least Square Regression* (LSR), FID requires less computation overhead and a smaller memory footprint, which makes FID more suitable to online model identification on resource-constrained wireless sensors. While communication overhead involved in FID includes those probe packets sent for collection of link quality data, such overhead is amortized by the long sampling period of FID.

V. Control Design

This section describes the control design of the P-TPC algorithm. We will first describe the design of a *Proportional-Integral Antiwindup* (PI-AW) controller, a key component of P-TPC . We will then provide the stability analysis of the proposed controller.

A. Design Goals

Our feedback transmission power control algorithm is designed to meet two primary requirements: (1) to maintain link quality, and (2) to reduce transmission power. Due to the uncertainties associated with low-power wireless communication, P-TPC adopts a feedback control approach that dynamically controls the link quality, specifically PRR, with power efficiency. Users specify a PRR set point y_R as the control objective. As an on-line power control algorithm, P-TPC is designed to enforce the PRR set point even in dynamic indoor environments, in which the multi-path effect and moving objects may undermine the link quality. Moreover, P-TPC is specifically designed to accommodate the motes' underlying hardware limitations, such as limits on the radio's transmission power and stringent constraints on memory consumption and computational capability.

B. PI-AW Controller Design

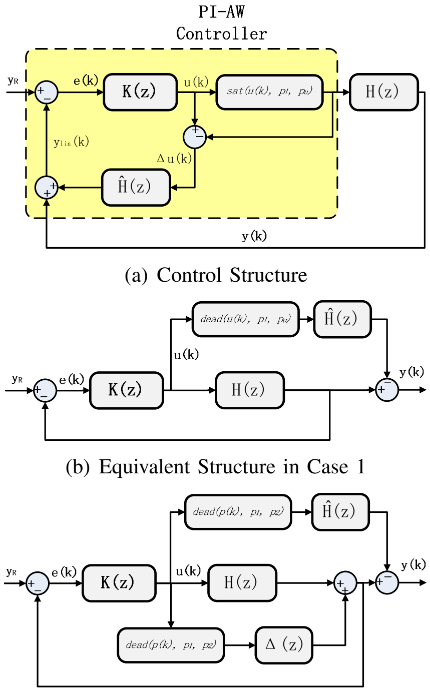
To enforce the set point subject to the actuator saturation, which is governed by the power bounds p_l, p_u, a PI-AW controller is designed [15]. PI-AW extends the classical PI controller to handle bounds on controller outputs (i.e., the upper and lower bounds of transmission power in our system). Figure 2(a) illustrates the structure of the proposed PI-AW controller. Our design consists of a PI controller (denoted as $K(z)$), an anti-windup controller (denoted as $\hat{H}(z)$) determined from the link model $H(z)$, and a saturation block, $u(k) = sat(\bar{u}(k), p_l, p_u)$, in which

$$sat(x, x_{\min}, x_{\max}) = \begin{cases} x_{\min}, & \text{if } x < x_{\min} \\ x_{\max}, & \text{if } x > x_{\max} \\ x, & \text{otherwise,} \end{cases}$$

where z represents $z - transform$ [11]. The PI controller's output is limited by the saturated block: i.e., the controller output cannot surpass the power bound indicated by the model. Essentially, the anti-windup controller transforms the nonlinear behavior of link quality induced by the power bounds to linear behavior, so that normal linear control design can be exploited.

The input of the PI controller is the error between the actual PRR and linearized PRR $y_{lin}(k)$. The control output of the PI controller, $\bar{u}(k)$, is limited by the saturated block to enforce power bounds. The error between $\bar{u}(k)$ and $u(k)$, denoted as $\Delta u(k)$, is passed through a link model between a sender-receiver pair (denoted $\hat{H}(z)$) which generates a compensation term $\Delta y(k)$; when combined with the actual PRR $y(k)$, a *linearized* PRR difference, $y_{lin}(k) = y(k) + \Delta y(k)$, is fed back to the controller $K(z)$ in order to guarantee stability. This compensation is also known as *anti-windup* control [15].

To design the controller with the proposed structure, we follow two steps. First, a nominal linear controller $K(z)$



(a) Control Structure

(b) Equivalent Structure in Case 1

(c) Equivalent Structure in Case 2

Fig. 2. Control Structure and Its Equivalent Structure

ignoring the saturation is designed. In this work, the nominal linear controller is a PI-controller

$$K(z) = K_{\mathrm{P}} + \frac{K_{\mathrm{I}}}{z-1}$$

Secondly, an anti-windup controller $\hat{H}(z)$ is designed to remedy performance deterioration when the power bounds are activated. For designing anti-windup control, we rewrite the model (6) in z-domain as follows:

$$H(z) = \frac{P(z)}{U(z)} = g, \tag{7}$$

where $P(z), U(z)$ are z transforms of $p(k), u(k)$.

Algorithm 1 presents psuedo-code of the PI-AW algorithm. The derivation of parameters K_I and K_P are given in the following section.

C. Stability Analysis

This section analyzes the stability of the proposed control framework. For a link between sender-receiver pair, stability ensures that the PRR converges to the set point if P-TPC is applied. From Figure 2(a), there are only two cases to maintain stability.

1) When the control input $p_l \leq u(k) \leq p_u$ (which implies that $\Delta u(k) = 0$) we want to enforce stability of the *active* closed-loop system consisting of $K(z)$ and $H(z)$, and stability of $\hat{H}(z)$.
2) When the control input saturates $u(k) \leq p_l$ or $u(k) \geq p_u$, we want to enforce stability of the *active* closed-loop system consisting of $K(z)$, $\hat{H}(z)$ and $\Delta(z)$ and the stability of $H(z)$.

Algorithm 1: PI-AW Controller

```
Input: PRR, $y(k)$
Output: Power level, $u(k)$
Data: PRR set point $y_R$; Power bound $p_l, p_u$; Power gain
      $g$; Controller parameters, $K_P$, $K_I$, $T_s$;
/* At the end of the $k^{th}$ sampling period                  */
$y_{lin}(k) = y(k) + \Delta y(k)$ /* linearized PRR            */
$e(k) = y_R - y_{lin}(k)$ ;
/* PI controller in velocity algorithm                         */
$\bar{u}(k) = u(k-1) + K_P(y(k-1) - y(k)) + K_I e(k)$ ;
if $p_l \leq \bar{u}(k) \leq p_u$ then
    $u(k) = \bar{u}(k)$;
else
    // Enforce power level constraints
    if $\bar{u}(k) < p_l$ then
        $u(k) = p_l$;
    else
        $u(k) = p_u$;
/* Anti-windup controller                                      */
$\Delta y(k+1) = g(\bar{u}(k) - u(k))$ ;
```

For the first case, as is assumed in [13], [15], stability of this system will first be considered for the special case in which $\hat{H}(z) = H(z)$. In this case it is straightforward to show that Figure 2(a) can be drawn in the equivalent form as depicted in Figure 2(b). The function $dead(u, p_l, p_u)$ is implemented as follows:

$$dead(u, p_l, p_u) = \begin{cases} (u - p_l), & \text{if } u \leq p_l \\ 0, & \text{if } p_l < u < p_u \\ (u - p_u), & \text{otherwise.} \end{cases}$$

The verification of stability of the closed-loop system in Figure 2(a) is based on the following well-known stability criteria in frequency domain.

Lemma 1: [11, p.857] Consider the closed loop consisting of $K(z)$ and $H(z)$ only depicted in Figure 2(b). In order for this loop to be stable, the net number of *counterclockwise* encirclements of the point -1 by the Nyquist plot of $K(e^{j\omega})H(e^{j\omega})$ as ω varies from 0 to 2π must *equal* the number of poles of $K(z)H(z)$ outside the unit circle.

From Lemma 1 and Figure 2(b), we obtain Theorem 1, which verifies stability of the our proposed control structure (Figure 2(a)) directly.

Theorem 1: The closed-loop system depicted in Figure 2(a), in which y_R is the input and y_{lin} is the output, is stable if:

1) $K(z)H(z)$ satisfy Lemma 1;
2) $\hat{H}(z) = H(z)$.

In addition, if the output $y(k)$ is to reach a steady-state output for a given input y_R, then $\hat{H}(z)$ should be stable.

For the second case we note that when

$$\hat{H}(z) = H(z)(1 + \Delta(z)),$$

Figure 2(a) can be shown to be in the equivalent form depicted in Figure 2(c). Therefore, when checking for stability, one should verify whether $K(z)\hat{H}(z)$ also satisfies the Nyquist stability criteria.

Theorem 1 and Lemma 1 lead us to the following theorem:

Theorem 2: The closed-loop system with controller

$$K(z) = K_\mathrm{P} + \frac{K_\mathrm{I}}{z-1}$$

depicted in Figure 2(a) is stable if:

1) $K_\mathrm{P} = \frac{K_\mathrm{I}}{2} = \frac{k_\mathrm{GM}}{2g_{\max}}$
2) $\hat{H}(z) = \hat{g}$, $\hat{g} \leq g_{\max}$

in which $k_\mathrm{GM} = 10^{-\frac{\mathrm{GM}}{20}}$, $g_{\max}$ is the upper bound of the power gain, and GM is the desired worst-case gain margin, which indicates that the system gain can be increased before the stability boundary is reached.

Proof: We give the proof of two claims separately.

1) Let us first consider the case of the closed loop system only with $K(z)$ and $H(z)$. The plant-controller loop-product can now be written in the following form:

$$K(z)H(z) = K_\mathrm{P}g + \frac{K_\mathrm{I}g}{z-1}. \tag{8}$$

The model of (8) indicates that no poles exist outside the unit circle for all $T_s < \infty$; therefore, Lemma 1 will always be satisfied if

$$|K(e^{j\pi})H(e^{j\pi})| \leq 1$$

This condition is sufficient that the phase margin will be greater than zero when $\omega = \pi$. It is therefore sufficient to let the magnitude of $k_{GM} = |K(e^{j\pi})H(e^{j\pi})| < 1$ or the magnitude of the respective proportional term (involving K_P) and integral term (involving K_I) to each be less than one-half when $\omega = \pi$ and can indeed be readily verified from our first expression given for $K(z)H(z)$. Thus we can choose K_I and K_P as follows to maintain the stability of the closed-loop system:

$$K_\mathrm{P} < \frac{k_\mathrm{GM}}{2g_{\max}} \leq \frac{k_\mathrm{GM}}{2g}$$

$$K_\mathrm{I} < \frac{k_\mathrm{GM}|(e^{j\pi}-1)|}{2g_{\max}} = \frac{k_\mathrm{GM}}{g_{\max}} \leq \frac{k_\mathrm{GM}}{g}.$$

2) According to the above proof, $K(z)$ is chosen to make the closed-loop system stable even g_{max} is adopted. Thus $|K(z)\hat{H}(z)| \leq |K(z)H(z)| < 1$, which means the closed loop involved $\hat{H}(z)$ is also stable. Therefore, based on Theorem 1, the whole system with anti-windup controller is also stable.

■

Theorem 2 provides a convenient method to design and reconfigure the PI-AW controller. According to Theorem 2, the parameters of the PI-AW controller are only related to k_{GM}, which is predefined by the users; and $g_{\max}$, which is upper bound of the slope of the linear model. Note that the parameters chosen by Theorem 2 are not optimal in sense of minimizing performance loss due to the disturbances. However, once the control algorithm is stable, it can offset most part of disturbance.

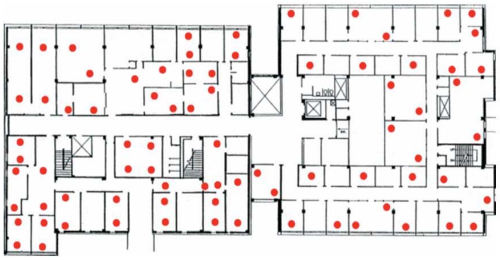


Fig. 3. Testbed in Bryan and Jolley Hall of Washington University in St. Louis. Red dots indicate the sensors.

D. Online Reconfiguration

By integrating efficient FID and PI-AW, P-TPC has an unique opportunity to reconfigure the controller online. Once the model parameters (p_l, y_l, p_u, y_u) are identified, PI-AW can update its parameters according to Theorem 2. Note that the design space of g is confined in $0 \leq g \leq 100$, since the power level is discrete and g is the ratio between PRR and power level. Hence, we can choose l *pivots*, $g^1, g^2, \ldots, g^l$, from the g design space as a predefined *pivot table* which is stored on the sensor. To design a controller for a model with the updated parameter g, we choose a *pivot* g^k from the *pivot table* and compute the parameters of PI-AW controller according to Theorem 2. The condition to choose the *pivot* g^k is that g^k is the closest but greater than g of the current link model according to the stability analysis. In our implementation, $l = 10$ and total 120 bytes are employed to store the table.

VI. Evaluation

To evaluate the proposed algorithms, we performed simulation based on the data trace collected from the testbed located in a campus building. And to further validate our approach, we implement the P-TPC in TinyOS 2.1.1, evaluate it on a testbed of TelosB motes, and compare its performance against three other representative TPC protocols.

A. Simulation And Experiments Setup

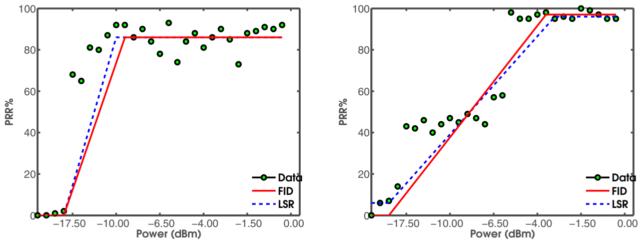
As the input of simulation, we collect data trace from a WSN testbed deployed in Bryan and Jolley Hall of Washington University in St. Louis. The WSN testbed consists of Tmote Sky and TelosB [21] motes, which are equipped with an IEEE 802.15.4 compliant Chipcon CC2420 radio [1]. The CC2420 can be programmed to operate on 31 power levels (numbered 1 to 31). We intentionally avoid two lowest power level settings, since the data sheet [3] shows the output power of transmitter using these two power levels are highly unstable.

Our experiments for collecting data trace are performed on top of the TinyOS 2.1 operating system [2] using the CC2420 driver's default CSMA/CA MAC layer. To measure the link quality of all power setting at a fine granularity, we deployed a single sender and multiple receivers in the experiments. This allows us to test multiple links simultaneously while avoiding interference between senders. In each run of experiment, the transmitter sent a batch of 100 consecutive packets to the broadcast address using a power level, then proceeded to the next power level in a round robin fashion. The process of sending 29 batches of 100 packets repeated every 5 minutes. The receiver nodes record the packet reception ratio (PRR), defined as the fraction of transmitted packets successfully received by the receiver.

We compare P-TPC against three baseline algorithms ART, ATPC and MPC. ART is a heuristic adaptive power control approach [14]. Based on fluctuations in link quality, ART increases/decreases power level one unit at each sampling period. ATPC is a feedback transmission power control algorithm based on RSSI measurement [18]. A simple linear RSSI model is identified offline; a variant of proportional control is employed online to change the power level. MPC is an optimization based control-theoretic algorithm which maintains the tradeoff between link quality and power efficiency by weight assignment to both parts in optimizing objectives [28]. Due to its optimality, we expect MPC to outperform P-TPC and ART. However, its expensive optimization computations prevent it from being implemented on resource constrained platforms such as the TelosB.

B. Model Identification

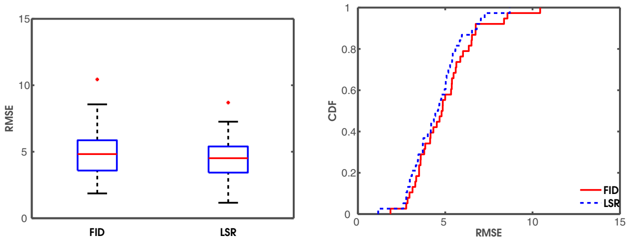
Before presenting P-TPC's performance, we first compare model identification of FID with the standard *Least Square Regression (LSR)* method. Fig. 4 shows the model identification results of FID and LSR for a single link at two specific times. On the one hand, both methods provide the piecewise linear relationship between transmission power and PRR which correctly identifies the transitional level between good and poor link quality. Although the two methods identify a slightly different corner point, the models that they derived are very similar.



(a) Result of Link 7 at 600 Min. (b) Result of Link 7 at 1250 Min.

Fig. 4. Model Identification Result of Link 7.

Fig. 5 compares the Root Mean Square Error (RMSE) of two methods over all links in our dataset, providing a comprehensive view of their identification results. As shown in Figure 4(a), FID and LSR achieve 4.9% and 4.8% median error in PRR, respectively, which indicates the average accuracy of FID quite close to that of LSR. Figure 4(b) plots the CDF of both models' error; although LSR generates solutions with less error than FID, the difference is again small. After closely examining the results, we attribute this difference to fluctuations in PRR caused by wireless interference. However, as we shown in the simulation and testbed results, PI-AW is robust against such disturbances even in the face of model uncertainties.



(a) Model Error (b) CDF of Model Error

Fig. 5. Comparison of Error of Model Identification Algorithms

To investigate the effects of interference on model parameters, we first compare the model identification results at different times for Link 7, as shown in Fig. 4. In Fig. 6, we plot the model identification result of (p_u, y_u), the turning point that the link quality changed from transitional area to the stable high area, for each 5 minutes during a long time interval. In both Fig. 4 and Fig. 6, we observe significant change in (p_u, y_u) which indicates non-negligible model variation. Specifically, from $100min$ to $700min$, y_p fluctuates in the range of $(25, 92)$ while p_u in the range of $(3, 30)$. Such significant parameters variation can not be handled by feedback control alone. Hence P-TPC employs online model identification and control algorithm reconfiguration to accommodate such dynamic and complex environments.

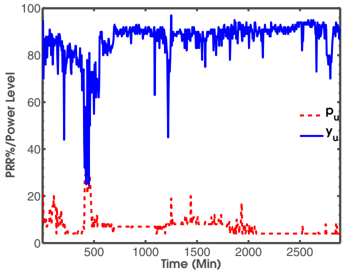


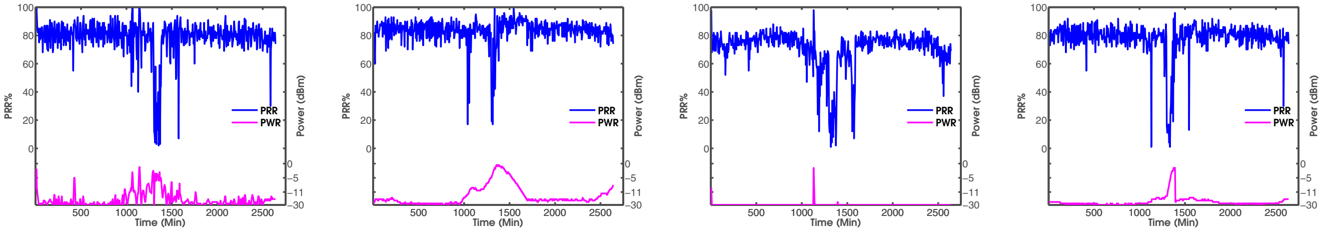
Fig. 6. Model Identification Result of (P_u, y_u) for Link 7.

C. Trace Simulation

In this section, we present simulation results based on data traces of a set of links collected from the WSN testbed. To probe the transmission power from different perspectives, we resent both direct transmission power data and the metric of Power Per successful Packet (PPP), defined as the power consumed to transmit one packet *successfully*.

1) Single Link Results: To understand the performance of all four protocols, we first look closely at their performance over a single link (link 17) from the dataset. For this simulation, all protocols' PRR set point are set to 80%.

Around 1300 min, external interference causes the PRR to drop significantly. P-TPC responds by rapidly increasing



(a) P-TPC (b) ART (c) ATPC (d) MPC

Fig. 7. Comparison of Simulation Results of Link 17.

transmission power to counteract the drop in link quality as shown in Fig. 7(a).

In contrast, as illustrated in Fig. 7(b), ART is significantly slower to react to link quality changes. After the link quality drop, ART spends around 400 minutes increasing the transmission power to the maximum, then slowly decreases it over the course of 300 minutes when the link recovers. This inefficiency can be attributed to ART's simple heuristic approach of adjusting the transmission power one unit at a time.

ATPC's performance is plotted in Fig. 7(c). We observe that ATPC only adapts the transmission power at a single point during the experiment, and does not adequately compensate for the drop in link quality. We attribute this behavior to deviation between RSSI and PRR: in the face of interference, RSSI can remain high in spite of a reduced PRR. This is consistent with observations in previous empirical study of link quality that RSSI is not a *consistent* index for link quality.

Finally, we plot MPC's performance in Fig. 7(d). Comparing Fig. 7(a) and 7(d), we observe that P-TPC and MPC generally have similar performance, with MPC achieving slightly better link quality. As discussed above, MPC achieves this via an expensive optimization approach which cannot realistically be deployed on constrained hardware.

2) Overall Results: To acquire a comprehensive evalutation of performance, we plot the PRR of all links in the dataset in Fig. 8(a). The median PRR for P-TPC, ART, and MPC are within 2% of the 80% set point. ATPC has the largest variance of PRR while MPC has the smallest. As we explained previously, MPC aims to optimize the performance. Although it consumes more power than P-TPC , as shown in Fig. 8(b) and Fig. 8(c), it can achieve better link quality. Again the difference of link quality results between P-TPC and MPC is not significant. Link quality of ART is close to P-TPC and MPC. However, as shown in Fig. 8(b) and Fig. 8(c), its power consumption is larger than that of P-TPC and MPC for almost all links because P-TPC and MPC adopt more aggressive power control mechanism.

D. Implementation

To evaluate P-TPC's real-world performance, we implemented the protocol in TinyOS 2.1.1 [2] and performed a data collection benchmark on our testbed as shown in Fig 3. The experimental setup is the same as the one used to gather the data traces above. However, rather than collecting data on all power setting, we deployed a full implementation of P-TPC below our application logic and allowed P-TPC to automatically select the transmission power.

	ROM (bytes)	RAM (bytes)
ART	17504	1534
P-TPC	22406	1782

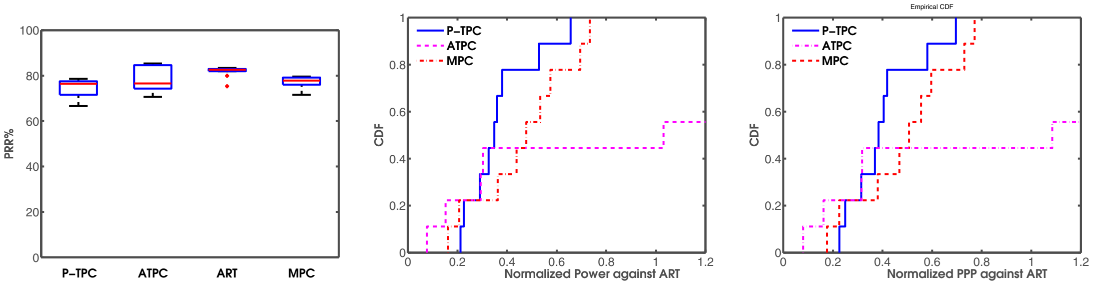
TABLE II
ROM AND RAM USAGE COMPARISON OF OUR BENCHMARK APPLICATION.

Owing to its lightweight design, P-TPC introduces very little memory overhead. Table II shows the program ROM and RAM usage for the benchmark application as reported by the TinyOS toolchain. Compiling the application with P-TPC enabled only consumes 3747 extra bytes of ROM and 208 bytes of RAM compared to the same application compiled with a fixed channel assignment. (For comparison, TelosB has 48 kilobytes of ROM and 10 kilobytes of RAM.)

E. Testbed Experiments

We selected 24 links at random from the 524 links detected in our testbed and performed the 24-hour experiments twice: once with P-TPC and once with ART. The two experiments were performed in two consecutive workdays to evaluate the impacts of different network conditions. We use a data rate of 20 packet per minute. This configuration emulates the behavior of typical environment monitoring applications, which feature relatively low data rates but require reliable data delivery. To simulate the case of real WSN applications, we set the PRR set point as 80% since [24] suggests that links with a PRR below 10% were found to be poor-quality, and links with a PRR between 10% and 80% to be bursty. Accordingly, we use a PRR of 80% throughout this section as a threshold to designate links as *good* or *reliable*.

1) Single Link Behavior: Fig 9 shows the experiment result of running P-TPC on Link 15 more than 100 hours. Before 3000 min, the transmission power is around 0 dBm and the PRR maintains 80%. After 3000 min, the link behavior changes significantly. The rule-based FID is activated to identify a set of updated model parameters, by which the controller is redesigned. At this stage, the transmission power is only around -5 dBm while the PRR is greater than 90%. After



(a) PRR (b) Power CDF (c) PPP CDF

Fig. 8. Comparison of Simulation Results for All Links in Testbed

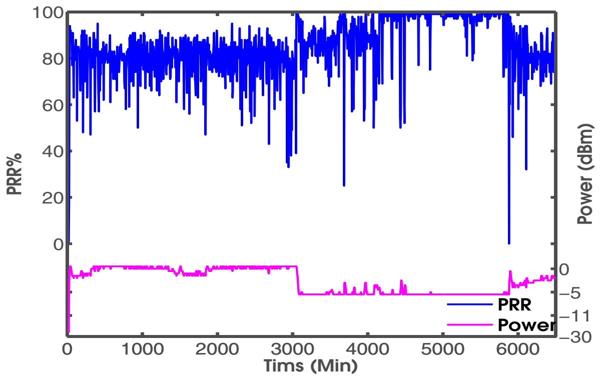


Fig. 9. Experiment Results of Link 15.

6000 min, model transition happens again. The PRR maintains 80% while the transmission adapts to enforce the PRR set point.

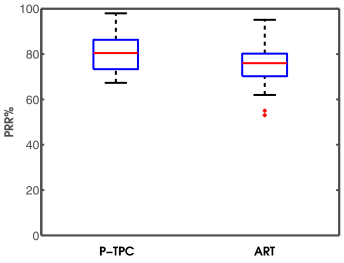
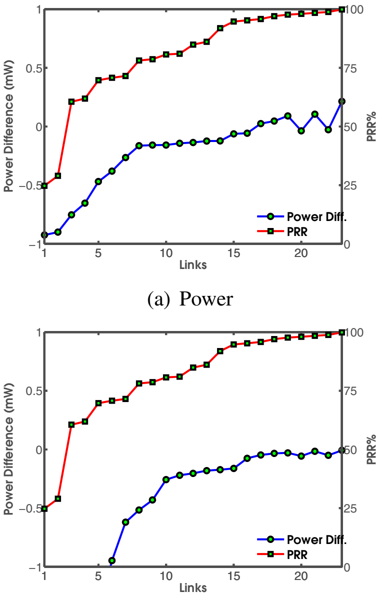


Fig. 10. Experiment Results of PRR under P-TPC vs. ART

2) Overall Performance: Fig. 10 compares the link quality under P-TPC and ART. Both P-TPC and ART maintain the PRR set point since the medians of PRR of two algorithm are close to 80% while P-TPC slightly outperforms ART because the median PRR under P-TPC is higher and closer to the set point than that of ART.

To investigate when P-TPC is more energy efficient than ART, we sort the results in the ascending order of PRR when without transmission power control. Fig 11(a) shows a clear trend of power difference: *the worse the link quality when without transmission power control, the more transmission power P-TPC saves when compare to ART*. For example, when the PRR of the link is 80% without transmission power control,



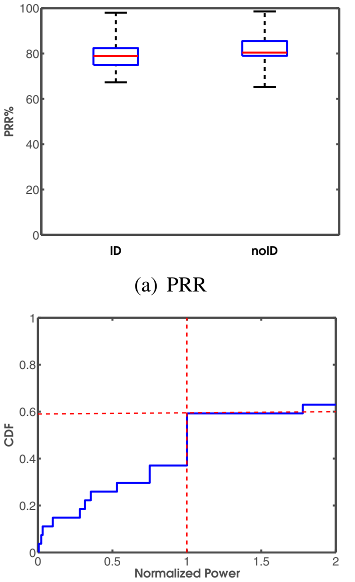
(a) Power

(b) PPP

Fig. 11. Sorted Power and PPP Difference between ART and P-TPC according to Ascending Order of PRR without Transmission Power Control

P-TPC consumes around less $0.2mW$ than that of ART. For those *good* links (whose $PRR > 80\%$), energy efficiency of the P-TPC is no worse than ART. In contrast for *poor* quality links, P-TPC consumes significantly less transmission power than ART.

3) Effect of Online Model Identification: As shown in Fig. 6, the power model of the link may experience significant variation, which motivates online model identification and controller reconfiguration. To examine how much the online model identification can improve performance, Fig. 12 compares the PRR and power of the same set of links running P-TPC with and without online model identification. In Fig. 12(a) we observe that while the median of PRR is almost the same in two cases, but online model identification reduces the variation of PRR. Fig. 12(b) shows the transmission power of P-TPC with online model identification normalized over that without online model identification. It is observed 59% links consume less transmission power if equipped online model identification than that of no model identification.



(a) PRR

(b) Normalized Power (Online model ID against No online model ID)

Fig. 12. Results of P-TPC with/without Online Model Identification

For highly dynamic and complex environments, model identification can improve the link quality and save transmission power as illustrated by above experimental results. However online model identification incurs communication overhead due to probe packets and computation overhead of online controller configuration. Hence whether it includes online model identification in P-TPC depends on tradeoff between overhead and performance.

VII. Conclusion

Transmission power control is an important approach to reduce the power consumption in wireless sensor networks. However, despite a multitude of existing protocols, they still face significant challenges in real-world deployments. This paper presents P-TPC , a practical TPC protocol designed based on control-theoretic techniques. P-TPC features a highly efficient controller designed based on a dynamic model that combines a theoretical link model with online parameter estimation. The robustness and energy savings under P-TPC are demonstrated through trace-driven simulations and real-world experiments in a campus building.

References

[1] 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, Texas Instruments.

[2] http://www.tinyos.net/.

[3] Cc2420 data sheet. In *http://www-inst.eecs.berkeley.edu/ cs150/Documents/CC2420.pdf.*

[4] S. M. M. Alavi, M. J. Walsh, and M. J. Hayes. Robust distributed active power control technique for IEEE 802.15.4 wireless sensor networks A quantitative feedback theory approach. *Control Engineering Practice*, 17(7):805–814, 2009.

[5] B. Zurita Ares, P. G. Park, C. Fischione, A. Speranzon, and K. H. Johansson. On Power Control for Wireless Sensor Networks: System Model, Middleware Component and Experimental Evaluation. In *ECC, 2007*, 2007.

[6] D. Blough, M. Leoncini, and G. Resta. The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *Mobile Computing, IEEE*, 5(9), 2006.

[7] D. Blough, M. Leoncini, and G. Resta. Topology control with better radio models: Implications for energy and multi-hop interference. *Performance Evaluation*, 64(5), 2007.

[8] C. A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig. The Impact of Temperature on Outdoor Industrial Sensornet Applications. *IEEE Transactions on Industrial Informatics*, 6(3):451–459, Aug. 2010.

[9] M. Burkhart and P. V. Rickenbach. Does topology control reduce interference? In *MobiHoc*, 2004.

[10] O. Chipara, G. Hackmann, C. Lu, W. D. Smart, and G.-C. Roman. Practical modeling and prediction of radio coverage of indoor sensor networks. In *IPSN*, 2010.

[11] G. F. Franklin, J. D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison Wesley Longman, Inc., 1998.

[12] Y. Gao, J. Hou, and H. Nguyen. Topology control for maintaining network connectivity and maximizing network capacity under the physical model. *INFOCOM 2008*, 2008.

[13] G. Grimm, A. Teel, and L. Zaccarian. The l2 anti-windup problem for discrete-time linear systems: Definition and solutions. *Systems & Control Letters*, 57(4):356–364, 2008.

[14] G. Hackmann, O. Chipara, and C. Lu. Robust topology control for indoor wireless sensor networks. In *SenSys*, page 57, New York, New York, USA, Nov. 2008. ACM Press.

[15] G. Herrmann, M. Turner, and I. Postlethwaite. Discrete-time and sampled-data anti-windup synthesis: stability and performance. *International Journal of Systems Science*, 37(2):91–113, 2006.

[16] L. Li, J. Halpern, P. Bahl, and Y. Wang. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *PODC*, 2001.

[17] N. Li, J. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, 2005.

[18] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. ATPC: adaptive transmission power control for wireless sensor networks. In *SenSys*, Oct. 2006.

[19] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. Stankovic, and T. He. Towards stable network performance in wireless sensor networks. In *RTSS'09*, pages 227–237. IEEE, 2009.

[20] N. Pantazis and D. Vergados. A survey on power control issues in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 9(4):86–107, 2007.

[21] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN*, 2005.

[22] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys (CSUR)*, 37(2):164–194, 2005.

[23] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *IEEE SECON*, 2004.

[24] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low power wireless. In *ACM Transactions on Sensor Networks*, 2010.

[25] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *SenSys*, page 29, New York, New York, USA, Nov. 2008. ACM Press.

[26] Theodore S. Rappaport. *Wireless Communications - Principles And Practice, 2/e*. Pearson Education, 2009.

[27] M. J. Walsh, S. M. M. Alavi, and M. J. Hayes. Practical assessment of hardware limitations on power aware wireless sensor networks-An anti-windup approach. *International Journal of Robust and Nonlinear Control*, 20(2):194–208, Jan. 2010.

[28] K. . Witheephanich and M. J. . Hayes. On the applicability of model predictive power control to an IEEE 802.15.4 wireless sensor network. In *ISSC 2009, IET Irish*, 2009.

[29] K. Witheephanich, J. Escaño, and M. Hayes. Explicitly constrained generalised predictive control strategies for power management in ambulatory wireless sensor network systems. In *ACC*, 2010.

[30] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *IEEE SECON*, 2004.