

Robust EKF-Based Wireless Congestion Control

Zhipeng Huang Xiaolong Li Homayoun Yousefi'zadeh

Department of EECS

University of California, Irvine

[zhipengh,xiaolonl,hyousefi]@uci.edu

Abstract—Many of the recently proposed high performance congestion control protocols rely on an estimation of the available link bandwidths. Specially, XCP and VCP need an explicit knowledge of the link bandwidth. However, wireless networks are characterized by bandwidth variations due to the openness of air links. Experiments have clearly shown that operating over variable bandwidth wireless links can lead to a significant performance degradation of congestion control protocols. Such degradation is typically manifested in the form of exhibiting oscillatory behavior. In this paper, we propose the use of an Extended Kalman Filter (EKF) to filter out the impact of bandwidth variations from the operation of wireless congestion control protocols. Our EKF-based Bandwidth Estimation (EBE) scheme can predict link capacity by monitoring the persistent queue size of a wireless link thereby eliminating the need for directly measuring the real-time bandwidth. We implement EBE in NS-2 and integrate it with XCP and VCP protocols. Through extensive simulation studies, we demonstrate significant performance improvements of both protocols in wireless networks as the result of using EBE.

Index Terms—Wireless Congestion Control, Time-Varying Bandwidth, Bandwidth Estimation, Extended Kalman Filter, Oscillatory Behavior, XCP, VCP, NS-2.

I. INTRODUCTION

Conventional congestion control protocols often fail to achieve both high bandwidth utilization and fairness as the result of relying on a binary packet loss signal to detect congestion. To address the problem, a wide variety of TCP variants such as those of [1], [2], [3] utilize an estimation of the available link capacity as well the Round Trip Time (RTT) to predict congestion. Beyond that, recently proposed eXplicit Congestion-control Protocol (XCP) [4] and Variable-structure Congestion-control Protocol (VCP) [5] rely on an explicit knowledge of the link capacity as an input to their more sophisticated congestion control algorithms. Both XCP and VCP are router-assisted protocols and need to compute Feedback or Load Factor (LF) to perform congestion control. While routers in XCP explicitly dictate to the sender the next sending rate, VCP routers only signal the congestion level derived from the link LF leaving the adjustment of the sending rate to the sender. Despite their distinctive designs, both protocols rely on an important measure, namely the link capacity, to perform congestion control.

Although link capacity never changes in wired networks, bandwidth variations happen frequently in wireless networks due to the openness of air links. Precise and accurate tracking of the actual capacity of wireless links is not a trivial task. Nonetheless, congestion control protocols such as XCP and

VCP cannot operate efficiently without an accurate estimation of wireless link capacity. The latter fact constitutes the main motivation for the work of this paper.

Bandwidth estimation techniques have been widely studied recently. Active probing techniques, such as those proposed in [6], [7], [8], [9], [10], mainly rely on actively injecting probing packets into the network. While these approaches perform reasonably well and on precision, they may exaggerate the congestion situation of an already congested network specially in high Bandwidth-Delay Product (BDP) networks. To the contrary, passive probing techniques such as those proposed in [11], [12], [13], [14], rely on passive measurement of the available bandwidth at the routers thereby avoiding the impact of packet injection. However, passive probing techniques typically suffer from the lack of precision in addition to requiring to apply changes that are some times major to the implementation of existing congestion control protocols.

In this paper, we propose a protocol-agnostic EKF-based bandwidth estimation scheme to which we refer as EBE. EBE can be used by any congestion control protocol for the purpose of precisely estimating the real-time link bandwidth. In this paper the phrases link bandwidth and link capacity are used interchangeably. In order to measure capacity in multi-rate open air media, EBE computes the persistent queue size locally at individual routers as oppose to using direct measurement of the capacity or using a packet-probing technique. Most notably, EBE utilizes an extended Kalman filter to provide an accurate estimation of the link capacity. Specifically, EBE computes two types of queue sizes: the average queue size and the estimated queue size. While the former is computed from the sampled queue sizes during the last control interval, the latter is provided by the EKF. The difference between the average queue size and the estimated queue size represents the difference between the input bandwidth defined as the input traffic rate and the link capacity.

We note that while the idea of using the difference between the input bandwidth and the link bandwidth is similar to that of XCP-b [11], the EKF-based estimation technique of EBE is distinctly different from that of XCP-b. To highlight the differences, XCP-b can only react to the occurrence of a bandwidth variation. Consequently, it may be too slow to react to changes in high delay networks. In contrast, EBE predicts the bandwidth variation pattern and reacts more rapidly and effectively to changes than XCP-b. In addition and in contrast to XCP-b, EBE is not specific to XCP operating over IEEE 802.11 MAC.

This paper makes several contributions. First, it proposes EBE a novel protocol-independent module that can be applied

to any congestion control protocol requiring a precise knowledge of the link bandwidth. Second, EBE offers precision performance comparable to that of end-to-end active probing schemes while avoiding the use of active packet injections. Third, we propose an alternative for the measurement of the capacity without having the need to change any of the existing congestion control protocols of interest to this study and thus minimize the deployment obstacles in contrast. The rest of the paper is organized as follows. In Section II, we review the fundamentals of XCP and VCP as congestion control protocols of interest. In Section III, we present the operating details of our proposed EBE scheme. The results of our experimental studies are provided in Section IV. Section V provides the conclusion to this work and Appendix A includes the operating principles of the EKF used in this study.

II. BACKGROUND

In this section, we review the fundamentals of XCP and VCP as the congestion control protocols of interest to this study.

A. Fundamentals of XCP

In order to cope with the TCP performance problem in high BDP networks, XCP decouples efficiency control from fairness control and uses explicit feedback to inform the sender of an XCP session about the degree of the network congestion such that sender can adjust its sending rate. XCP requires support from both end and intermediate nodes and places the congestion control state information in the packet header. Then information contains the sender's current congestion window size and round trip time estimation as well as the feedback information which is initialized by the sender and modified by the router. An XCP router computes the feedback in a certain control interval, and puts the feedback information in the packet header. The feedback is calculated through the following equation

$$\Phi = \alpha \cdot d \cdot S - \beta \cdot Q \quad (1)$$

where α and β are constant parameters. S represents the spare bandwidth and is calculated by subtracting the input bandwidth value from the link capacity value. Q is the persistent queue size and computed as the minimum queue size observed for an arriving packet in the last propagation delay. It is obvious that the values of link capacity and persistent queue size significantly impact the generation of feedback and the determination of network status by XCP. Since the link capacity is variable in wireless network, accurately identifying link capacity values is crucial for XCP to perform as well as it does in the wired network.

B. Fundamentals of VCP

Just like XCP, VCP decouples efficiency control from fairness control allowing it to be able to perform well in high BDP networks. However unlike XCP, VCP utilizes a much smaller number of bits in the packet header than XCP for encoding congestion information. VCP uses the two ECN bits that already exists in the IP header to convey the level of the network

congestion. With VCP, each router periodically calculates the LF classifying the level of the network congestion into three regions: low-load, high-load, and over-load. The formula for the computation of the LF is as follows

$$\rho_l = \frac{\lambda_l + \kappa_q \cdot \tilde{q}_l}{\gamma_l \cdot C_l \cdot t_\rho} \quad (2)$$

where λ_l is the amount of the input traffic during time interval t_ρ . The parameter κ_q controls how fast the persistent queue drains, C_l is the link capacity, γ_l is the target utilization rate, and \tilde{q}_l is the persistent queue size. After the router computes the LF, it encodes it to the ECN bits of the IP header and sends it back via ACK packets to the sender. Upon receiving the LF, the sender applies Multiplicative Increase (MI), Additive Increase (AI), or Multiplicative Decrease (MD) of the congestion window size corresponding to low-load, high-load, and over-load congestion levels. Just like XCP, the values of the link capacity and the persistent queue size are very important to VCP for accurate computation of the LF. Thus, it is also vital for VCP to have an efficient method of determining the variable link capacity in wireless networks.

III. EKF-BASED BANDWIDTH ESTIMATION

As noted above, many high performance congestion control protocols require the estimation of the available link bandwidth. While such estimated information is needed by the routers in router-assisted protocols, it is only used by the end nodes in end-to-end protocols. The EBE scheme proposed in this paper introduces a passive protocol-agnostic link capacity estimation method for use by congestion control protocols while minimizing the impact of change to such protocols. Rather than directly measuring the link capacity, EBE takes the router's persistent queue size or the end-node's perceived RTT as its input based on the characteristic of the utilizing protocol. While the former is usually used for router-assisted protocols such as XCP and VCP, the latter can be used for end-to-end congestion control protocols such as TCP BIC. In this paper, we focus on the former case as it represents a more complicated task in comparison to the latter case. Thus, our target protocols for using EBE include router-assisted protocols operating in variable bandwidth wireless networks. Specifically, the key components of our proposed EBE scheme are as follows.

A. EKF Predictor: An EKF is used to estimate and predict the real-time persistent queue size. The output of the EKF is delivered to the Link Capacity Monitor in order to retrieve the instant link capacity as presented below. At this stage, the parameters of the EKF have been tuned for producing a precise prediction.

B. Link Capacity Monitor: To feed the EKF, EBE keeps track of the average persistent queue size computed during the last control interval. With the average persistent queue size, EBE makes an estimation of the real-time persistent queue size. The difference between the measured average queue size and the output of EBE can be utilized as the representation of the spare link capacity. While the link capacity is not directly computed, for those protocols like VCP that require

TABLE I
THE TABLE OF EKF PARAMETERS.

Variable	Type	Value
x_{pre}	vector(2)	average queue size sampled last interval
x_{pos}	vector(2)	estimated queue size for the next interval
z	vector(2)	directly measured queue size
P	matrix(2x2)	covariance of the sampled queue sizes
Q	matrix(2x2)	$Q(1,1) = 10^{-5}$, $Q(2,2) = 10^{-5}$
R	matrix(2x2)	$R(1,1) = 10^{-5}$, $R(2,2) = 10^{-5}$

the knowledge of the link capacity, it can be retrieved from the summation of the input bandwidth and the spare link capacity.

C. Queue Maintainer: As presented above, EBE relies on the variations of persistent queue size to detect a change to the link bandwidth. However, the change of persistent queue size results from the change of the input bandwidth depending on the sending rate. As the link bandwidth drops, the queue builds up due to slower draining speeds and thus EBE can detect the decrease easily. In contrast, if the congestion control protocol has reached its steady-state status when the link bandwidth increases, the sending rate of each sender may remain relatively stable at a low value. Thus, there is not sufficient variation of the persistent queue size that can be detected by EBE. Unless there is an accidental variation in the persistent queue size, EBE is unable to detect the increase. Thus, EBE provides a queue maintainer that can detect the link bandwidth increase while maintaining a reasonable queue size. Note that the performance of the queue maintainer represents a tradeoff between responsiveness and performance.

In what follows, each of the key components of EBE are described in greater details. Appendix I includes a brief mathematical review of the operating principles of an EKF.

A. EKF Predictor

As introduced above, EBE utilizes an EKF to predict the real-time persistent queue size. There are two timers associated with the EKF predictor. The first timer to which we refer as the measurement interval timer is for sampling the persistent queue size and expires once every 10ms. The value of the measurement interval timer is set such that it can provide the underlying protocol with ample time to converge. The second timer to which we refer as the estimation interval timer is for making the estimation of the real-time persistent queue size and expires once every 200ms. During the estimation interval, all sampled queue sizes are simply stored in a set. After computing the mean and the covariance of the set of the sampled queue sizes, the mean and the covariance are delivered to the EKF as input parameters x_{pre} and P , respectively. The EKF then starts its first cycle, the prediction cycle, in which it gets the a priori state estimate x and the a priori covariance estimate P for the current estimation. Thereafter, the EKF makes a measurement z of the directly measured instant queue size. Finally, the EKF will produce the Kalman Gain, the a posterior state x_{pos} , and the a posteriori covariance P_{pos} . Table I shows the main parameters defined for the EKF to calculate the estimation of the persistent queue size. As illustrated in the table, x_{pre} , x_{pos} and z are defined as one dimensional vectors each containing two elements. All

the second elements are set to 0.0, while the first elements are set to the values demonstrated in the “Value” column. P , Q , and R are defined as two dimensional matrices. All of the diagonal elements of the matrices are set to values shown in the “Value” column while other elements are set to 0.0. The first element in x_{pos} is the estimated persistent queue size for the next interval and is required for the capacity calculation. Notably, the estimation of EKF is sensitive to the value of parameter Q . In our experiments, we set $Q = 10^{-5}$ yielding the most accurate estimation.

B. Link Capacity Monitor

Instead of calculating the difference between the link capacity and the input bandwidth, EBE derives the spare bandwidth from the queue variation on each estimation by setting the spare bandwidth to Δq . The value of Δq represents the difference between the persistent queue size estimated by the EKF and the average persistent queue size computed in the last estimation interval. Further, EBE retrieves the link capacity by adding up the input bandwidth and the spare bandwidth. While EBE appears to take a similar approach as that of XCP-b in terms of monitoring queue dynamics, EBE differentiates from XCP-B in following aspects. First, XCP-b is specifically designed for XCP while EBE is designed to operate as a module that can potentially operate with any congestion control protocol requiring the knowledge of link capacity and/or spare bandwidth. While not presented in this paper due to the limitation of space, EBE can be run at the end nodes in order to provide end-to-end TCP variants with a spare bandwidth estimation using RTT dynamics or ACK arrival dynamics. Thus, EBE provides an open interface to any congestion control protocol. Second, EBE has a greater potential in compatibility in contrast to XCP-b. In our simulations, EBE provides the congestion controls with which it interacts an additional set of parameters on demand besides the input bandwidth and the persistent queue size. The set of the parameters, for example, could include packet arrival rate and persistent queue draining rate. Such a design potentially allows for an easier integration with many other protocols that take advantage of other network characteristics to estimate the link capacity. Most importantly, the output of EBE is a precise prediction rather than a direct measurement of XCP-b. Such a prediction allows the sender to react pro-actively to a potential link capacity change as oppose to reactively thereby being able to make appropriate adjustments before the change happens. Furthermore, in rapidly changing BDP networks, the prediction can achieve good responsiveness.

C. Queue Maintainer

The queue maintainer component is designed to increase the responsiveness of EBE in the case of bandwidth increase. Specifically, when link bandwidth increases, the persistent queue size is supposed to decrease. However, if the previous persistent queue size remains low or nearly zero, such a bandwidth increase will not affect the queue size in a visible degree. As a result, EBE may lack of the signal necessary to drive the EKF for making an accurate prediction. Thus,

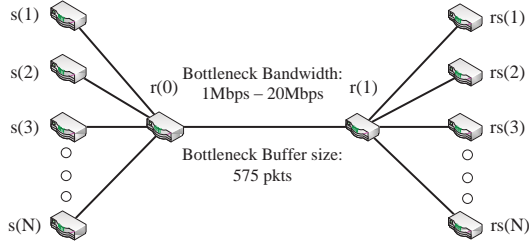


Fig. 1. The dumbbell topology used in our simulations.

the queue maintainer is introduced to force an increase in the sending rate and to build a reasonable value for the persistent queue. Specifically, the queue maintainer monitors the ratio of the average queue size in the current control interval and in the last control interval. We maintain a queue length that equals to 20% of the queue buffer size in our simulation. Therefore, for instance, if the ratio is smaller than or equal to 1.0 and both two average queue sizes are below the threshold value set to be 20% of the buffer size, the queue maintainer assumes the link bandwidth has increased. It, then, outputs a factor of Υ to the congestion control protocol which will yield an *inflated* load factor or feedback, until it detects the ratio is larger than 1.0 and the average queue sizes are larger than the threshold value. The choice of the Υ will vary slightly depending on the choice of the protocol. We set Υ to 1.14 for VCP and to 1.40 for XCP indicating our best experimental findings. Note that the choice of the value of Υ represents a tradeoff. A high value for Υ yields better bandwidth efficiency and a higher persistent queue value, while a low value for Υ yields a low speed of converge and a smaller persistent queue size. Notably, such a design does not require any support from the congestion control protocol itself.

At the end of this section, it is important to note that the queue maintainer is designed only for router-assisted congestion control protocols, as the adjustment of the sending rate is directly related to the feedback from routers. In order to estimate the available bandwidth for end-to-end protocols, EBE relies on monitoring the end-node perceived RTTs or ACK-arrival rates. As a sender keeps probing the available bandwidth by increasing its sending rate in different scales, the EKF can always vary its input and thus has no need for a queue maintainer.

IV. PERFORMANCE EVALUATION

In this section, our NS-2 performance profiling results are presented. We implement EBE in NS-2 simulator and apply EBE to both XCP and VCP. We demonstrate performance improvements of both protocols measured in terms of bandwidth utilization as the result of using EBE while also investigating the buffer occupancy measured in terms of the queue length. In our experiments, we refer to EBE improved XCP as XCP-EBE, and EBE improved VCP as VCP-EBE. In the case of XCP, we also compare our results with those of XCP-b.

Fig. 1 shows the dumbbell topology of our simulation study. The bottleneck link bandwidth is assumed to be varying randomly according to a uniform distribution.

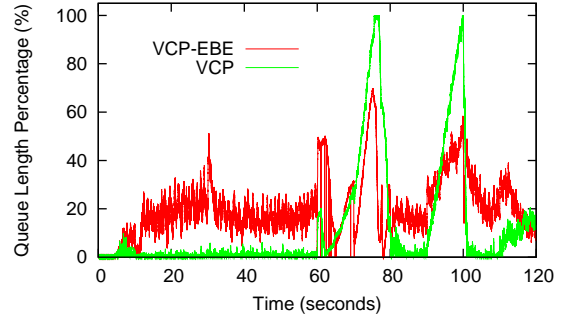


Fig. 2. A comparison of the persistent queue size of the bottleneck link for two variants of VCP. The buffer size is set to 575 KBytes.

All links are configured to have a one way delay of 4ms except the bottleneck link which is set to have a one way delay of 32ms. The bottleneck link bandwidth varies between 1Mbps and 20Mbps. A single simulation experiment has an overall duration of 120s.

There are two types of end-to-end aggregate FTP flows traversing over the topology, namely, the long-lived and short-lived flows. All flows represent two-way traffic. While the former consists of 20 FTP flows traversing the bottleneck links in both directions, the latter represents 10 FTP flows traversing the bottleneck link in the forward direction going from left to right. All flows begin at a random start time between 1ms to 300ms. The average packet size is 1 KBytes. The buffer size of the bottleneck link is set to 575 KBytes.

First we compare the performance of VCP and VCP-EBE. Then, the performance of XCP, XCP-b, and XCP-EBE are compared. For both XCP and VCP, the parameter *CAPACITY* is set to 10Mbps.

A. A Performance Comparison of VCP and VCP-EBE

Fig. 2 compares the persistent queue size of the bottleneck link in the case of VCP and VCP-EBE.

In the case of VCP, the behavior of the persistent queue size is quite different when the link capacity is larger or smaller than the configured capacity. While VCP successfully maintains a low queue size in the former case, oscillations are observed in the latter case. Essentially, VCP is unaware of the underlying bandwidth variations and makes adjustment of its sending rate using the fixed nominal value of the link capacity. Accordingly, VCP does not make any specific adjustment as the link bandwidth increases and therefore maintains a low queue size as usual. However as the link bandwidth drops, VCP still attempts to achieve a 10Mbps bandwidth throughput. Thus, the queue size increases and eventually overflows. Then, VCP reacts to overflow with a sending rate drop causing oscillation. In contrast, VCP-EBE can detect the change of the link bandwidth and respond appropriately. Over the entire simulation period, VCP-EBE maintains a relatively steady queue size, which is about 20% of the buffer size in our simulation. As illustrated below, although the average queue size for VCP-EBE is larger than that of VCP, the average queue size for VCP-EBE is in an acceptable range and shows the tradeoff between efficiency and buffer occupancy.

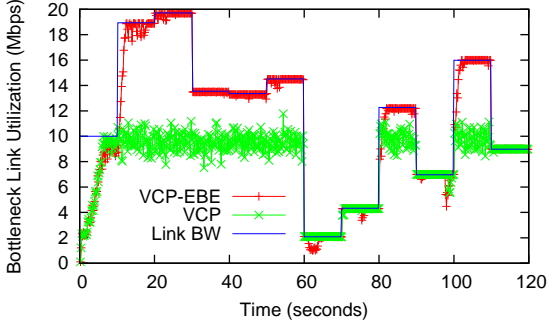


Fig. 3. A comparison of the bottleneck link utilization of VCP and VCP-EBE.

Fig. 3 compares the achieved bottleneck link utilization by VCP and VCP-EBE.

While the actual link bandwidth is larger than the configured value of 10Mbps in the time interval from 10s to 60s, VCP fails to utilize the increased bandwidth efficiently. When the bandwidth drops, VCP still keeps increasing its sending rate in order to achieve the perceived utilization calculated using the fixed bandwidth value. The latter results in growing the queue size of the bottleneck link and eventually making the link congested.

To the contrary, VCP-EBE demonstrates near 100% utilization during the whole simulation period illustrating good sensitivity and responsiveness to bandwidth variations.

B. A Performance Comparison of XCP, XCP-b, and XCP-EBE

In this section, we compare the performance of XCP, XCP-b, and XCP-EBE.

Fig. 4 compares the persistent queue size of the bottleneck link for the three variants of XCP. In the case of XCP and similar to the case of VCP, the behavior of the persistent queue size is quite different when the link capacity is larger or smaller than the configured capacity. XCP could successfully maintain a low queue size in the former case, while oscillations are observed in the latter case based on the same reasons explained in the case of VCP. Both XCP-b and XCP-EBE show similar queue size characteristics. While both schemes maintain a relatively steady queue size, their average queue sizes are larger than that of XCP over the entire simulation period. It is also observed that the average queue size of XCP-EBE is typically larger than that of XCP-b.

Fig. 5 compares the bottleneck link bandwidth utilization achieved by XCP, XCP-b, and XCP-EBE. A close look at the results reveals the following observations. First, XCP fails to utilize the increased bandwidth efficiently when the actual link bandwidth is larger than the configured value. When the bandwidth drops, XCP “blindly” increases the sending rate eventually causing a significant packet loss. To the contrary, XCP-EBE demonstrates near 100% utilization during the whole simulation period illustrating good sensitivity and responsiveness to bandwidth variations. XCP-b shows a similar behavior compared to XCP-EBE. However, XCP-EBE typically achieves a higher utilization and has a faster response when the bandwidth varies.

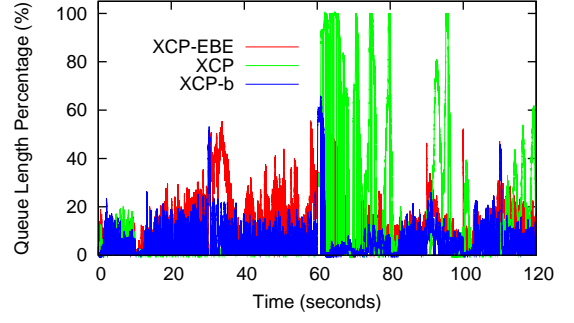


Fig. 4. A comparison of the persistent queue size of the bottleneck link for three variants of XCP. The buffer size is set to 575 KBytes.

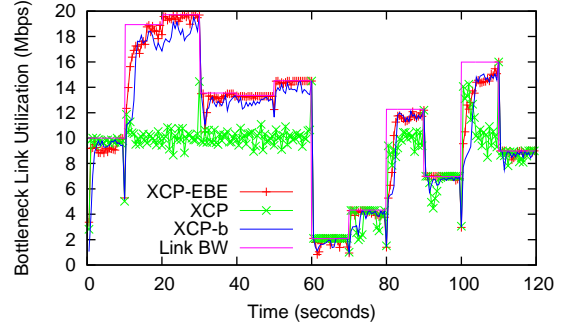


Fig. 5. A comparison of the bottleneck link utilization of XCP, XCP-b, and XCP-EBE.

V. CONCLUSION

In this paper, we proposed an Extended Kalman Filter based bandwidth estimation module that provided a high precision measure of link capacity for use by congestion control protocols. We demonstrated how our proposed protocol-independent non-packet-probing module is distinct from other comparable bandwidth estimation approaches. Rather than directly measuring the link capacity, our module used the persistent queue size and the associated parameters to calculate link capacity. Furthermore, our module could overcome the degredation of the network utilization problem resulted by an increasing bandwidth value. We implement our module in NS-2. Through simulation studies, we demonstrated that both XCP and VCP with EBE could overcome the limitations of the original protocols and achieve significant performance improvements in network utilization by eliminating the protocol oscillatory behavior.

APPENDIX A

THE OPERATING PRINCIPLES OF AN EKF

The mathematical representation of an EKF consists of a set of recursive equations that attempt at estimating the state of a process by minimizing the mean square error. Fig. 6, reproduced from [15], illustrates the high level functional view of an EKF. The EKF propagates the probability distribution function of the state variable and its covariance from a time instance to another. We refer the interested reader to [16] for related details. The EKF addresses the general problem of estimating the state $x \in \mathcal{R}^n$, of a discrete-time controlled

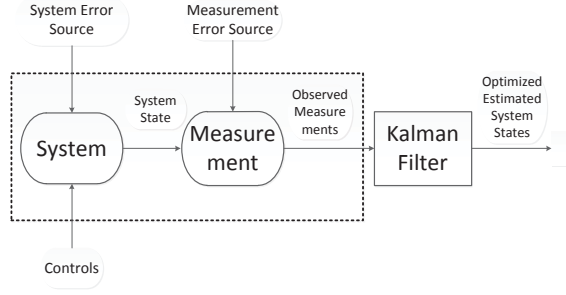


Fig. 6. A high level function view of an EKF.

process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3)$$

In the equation above, u is the control input and w is the process noise. The parameter w is assumed to have a normal probability distribution $N(0, Q)$, where Q is the process noise covariance matrix. In practice, Q may change from step to step but is assumed to be constant in this representation. A is defined as the state transfer function mapping the a priori state to the a posteriori state. Measurements that help the EKF to gain an accurate estimate, $z \in \mathcal{R}^m$, depend linearly on the system state vector as

$$z_k = Hx_k + v_k \quad (4)$$

In the equation above, the parameter v is assumed to represent the measurement noise and have a normal probability distribution $N(0, R)$, where R is the measurement noise covariance matrix. Once more, R may change from step by step in practice but is assumed to be constant here. H is defined as the measurement transfer function mapping the a posteriori state to the estimated state measurement.

The EKF estimates the state of a process by using feedback control. The filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. The operation of an EKF actually contains two cycles. The first cycle also known as the “prediction” cycle, projecting the current state and error covariance forward in time, obtains a priori estimates for the next time step. We define the a priori state estimate \hat{x}_k^- and the a posteriori state estimate \hat{x}_k , with the latter also taking the measurement z_k at step k into account. The second cycle also known as the “update” cycle, updates equations as the result of receiving feedback. In essence, the second cycle incorporates a new measurement into the a priori estimate in order to obtain an improved a posteriori estimate. This step is realized in the following equation

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (5)$$

where the Kalman gain K_k is given by

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (6)$$

In summary, the Kalman filter equations are given by

$$\begin{aligned} \hat{x}_k^- &= Ax_{k-1}^- + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \\ K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (1 - K_k H)P_k^- \end{aligned} \quad (7)$$

REFERENCES

- [1] L. Xu, K. Harfoush, and I. Rhee, “Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks,” in *Proc. of the Infocom 04*, 2004.
- [2] I. Rhee and L. Xu, “CUBIC: A New TCP-Friendly High-Speed TCP Variant,” in *Proc. of the PFLDNet’05*, Feb. 2005.
- [3] S. Floyd, “HighSpeed TCP for Large Congestion Windows,” Aug. 2002.
- [4] D. Katabi, M. Handley, and C. Rohrs, “Congestion Control for High Bandwidth-Delay Product Networks,” in *Proc. ACM SIGCOMM*, 2002, Aug. 2002.
- [5] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, “One More Bit Is Enough,” in *Proc. ACM SIGCOMM*, 2005, Aug. 2005.
- [6] K. Lai and M. Backer, “Measuring Link Bandwidth Using a Deterministic Model of Packet Delay,” in *Proc. ACM SIGCOMM*, Sept. 2000, pp. 283 – 294.
- [7] J. Bolot, “Characterizing End-to-End Packet Delay and Loss in the Internet,” in *Proc. ACM SIGCOMM*, 1993, pp. 289 – 298.
- [8] M. Jain and C. Dovrolis, “End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,” in *Proc. ACM SIGCOMM*, Aug. 2002, pp. 295 – 308.
- [9] B. Melander, M. Bjorkman, and P. Gunningberg, “A New End-to-End Probing and Analysis method for Estimating Bandwidth Bottlenecks,” vol. 1, pp. 415 – 420, 2000.
- [10] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman, “Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering,” pp. 73 – 84, Apr. 2006.
- [11] F. Abrantes and M. Ricardo, “XCP for shared-Access Multi-Rate Media,” vol. 36, no. 3, pp. 27 – 38, July 2006.
- [12] Y. Su and T. Gross, “WXCP: Explicit Congestion Control for Wireless Multi-hop Networks,” pp. 313 – 326, June 2005, available at <http://www.springerlink.com/content/xqvhwmdtj0bym6j/f/>.
- [13] S. Rangwala, A. Jindal, K. Jang, and K. Psounis, “Understanding Congestion Control in Multi-hop Wireless Mesh Networks,” pp. 291 – 302, 2008.
- [14] K. Chen and K. Nahrstedt, “EXACT: An Explicit Rate-based Flow Control Framework in MANET,” July 2002, available at <http://cairo.cs.uiuc.edu/projects/adhoc/exact.html>.
- [15] P. Maybeck, “The Kalman Filter: An Introduction to Concepts,” 1990.
- [16] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” July 2006, available at <http://www.cs.unc.edu/welch/kalman/kalmanIntro.html>.