Clock Synchronization Using Maximal Margin Estimation

Dani E. Pinkovich and Nahum Shimkin

***Abstract*— Clock synchronization in a network is a crucial problem due to the wide use of networks with simple nodes, such as the internet, wireless sensor networks and Ad Hoc networks. We present novel algorithms for synchronization of pairs of clocks based on Maximum Margin Estimation of the offset and skew between pairs of clocks. Our algorithms are inspired by the well known Support Vector Machines algorithm from the Machine Learning literature and have sound geometrical intuition for our model. In addition, we provide a modification to our algorithms (also relevant for the existing LP algorithm) to enhance their robustness to measurement outliers. Finally, we analytically derive the Mean Square Error for the estimation of offset, in the special case when the skew is given. Simulation experiments demonstrate that our algorithms have significantly better performance than state of the art synchronization algorithms.**

I. Introduction

Synchronization between pairs of clocks in a network is a very important task which has been treated extensively in the literature. Synchronization has specific standards such as the IEEE 1588 standard PTP [1] for LAN, specifically used for networked measurement and control systems. Specific protocols are used, the most prevalent of which is NTP [2]. In Wireless Sensor Networks (WSN), where multiple sensors observe parts of the same phenomenon and communicate over wireless protocols, synchronization is crucial to process the measurements correctly. See [3] for a comprehensive review. In these problems each computer (or node) has its own clock, where different clocks may differ in their current time indication (time offset), as well as in their frequency rate (skew).

Skew estimation can be performed using one-directional communication. In this model one clock sends its neighbor time-stamped messages. The second clock then measures its own time upon receiving these messages. To estimate both the offset and the skew bidirectional communication between the clocks is required. In this model the delays measured in both directions provide immunity against the constant network propagation delay. To estimate the offset and skew using bidirectional measurements between the clocks the two groups of measurements (outgoing and incoming messages) have to be separated by the line which estimates the offset and skew in the most accurate way. The bidirectional Linear Programming (BLP) presented in [4] estimates this line by finding two separate lines, one bounding the outgoing messages from below and one bounding the incoming messages from above, such that the sum of vertical distances between the line and all the measurement points is minimal. In Machine Learning, Support Vector Machines (SVM) [5] are used to separate between two classes, leading to state of the art classification results, see [6]. Inspired by SVM, we choose to estimate the offset and skew using Maximum Margin estimation. We seek for the two parallel lines farthest from each other which lie beneath all the points representing the outgoing communication and above all the points representing the incoming communication. We use simulations to show our method provides much more accurate synchronization results compared to state of the art methods, and brings the performance a step closer to the CRLB (Cramer Rao Lower Bound).

Previous Work: The Network Time Protocol (NTP) was presented in [2] and is today the standard protocol in the Internet. It performs online clock synchronization in a network, each measurement updating the skew and offset estimations of all the neighbors which receive the synchronization messages. However, high noise variations across networks make individual measurements very prone to significant errors. For this reason, real-life protocols such as NTP have gradually evolved over the years to obtain filters which allow the algorithm to neglect increasingly more noisy measurements. On the other hand, batch synchronization algorithms exist, which process a large set of measurements at once, incorporating different types of robustness in the algorithm. In [7] and [8] Paxson uses a robust line fitting technique to decrease the influence of the changing delay between measurements which introduces a lot of noise to the delay measurements. In [9], Moon, Skelly and Towsley dealt with one-way measurements. Their solution is a Linear Program which finds the line which lies beneath all the measurement points and has the lowest sum of vertical distances to all the points. The linear programming algorithm was compared to Paxson's algorithm in [9] and provided better results. Later in [10] it was shown that the linear program coincides with the Maximum Likelihood estimator for the one-way measurements case with additive i.i.d. exponential noise and unknown delay. In [4] Bletsas proposed to use bidirectional communication and solve two independent Linear Programs for the outgoing and the incoming messages. The average of the results comprises the algorithm's estimation. It was shown that the bidirectional LP algorithm (now not an ML estimator due to the change of setting) had superior performance to the Kalman filter and Average Time Difference. A large body of work has been devoted to finding Maximum Likelihood estimators for clock offset and skew in the two-way measurements case with exponential i.i.d noise, see

MED2011, June 2011

The authors are with Department of Electrical Engineering, Technion—Institute of Technology, Haifa 32000, Israel danipi@tx.technion.ac.il, shimkin@ee.technion.ac.il

[11], [12], [13] and [14]. Finally, in [15] the presentation is complete with ML estimators for known and unknown constant delay. However, the resulting algorithms have very high complexity.

For further details refer to our full technical report [16] and to the review paper [3].

Paper Overview: We begin by presenting the problem formulation and existing LP algorithms in section II. This forms the basis for the understanding of our Max Margin algorithms presented in section III. Then, in section IV we provide initial analysis of the offset estimation error of our algorithms and the LP algorithm. In section V we then show how it is possible to make our algorithms, and the Linear Programming algorithm robust to negative outliers. Section VI discusses the basic properties of the Max Margin algorithms we have presented. Finally, section VII shows simulations of the presented algorithms and section VIII concludes the article and discusses our future work.

II. Model Formulation and Existing LP Algorithms

Consider two clocks C_1 and C_2, situated in distant locations and connected by a (possibly wireless) network. Assume the first clock is a reference clock and we would like to estimate the offset and skew of the second clock relative to the first one. When the first clock shows the time $C_1(t) = t$, the second clock shows the time $C_2(t) = st + o$, assuming o and s are constants, i.e. that the clocks have no frequency drift.

A. One Directional Communication

We adapt the noise model of [10]. In this model we assume that clock C_1 sends L "outgoing" messages to clock C_2 at C_1's times $t_1, \ldots, t_L$. Each message sent at time t_l reaches C_2 at $t_l + d$ where d is the unknown constant part of the propagation time in the network between the two clocks. Clock C_2 then shows its time:

$$y_l = C_2(t_l + d) = st_l + d + o + \epsilon_l \tag{1}$$

where ϵ_l represents the variable portion of the propagation delay and the measurement noise. This noise has been modeled in the literature to be distributed mostly as Exponential, but also as Gaussian, Gamma and Weibull, see [17], [18] and [19]. We too model it as an exponentially distributed random variable with mean β, $\epsilon_l \sim \mathrm{Exp}(\beta^{-1})$, $p_{\epsilon_l}(x) = \beta^{-1}e^{-\beta^{-1}x}, x \geq 0$. The noise distribution is one-sided since the messages can only arrive after being sent and having traversed the network. See figure 1 for an illustration of the one directional measurement model.

The Linear Programming algorithm to estimate the offset and skew in one directional communication was proposed in [9]. The geometric intuition behind this algorithm is finding the straight line which lies beneath all the measurement points in the (t, y) plane, and has the lowest sum of vertical distances to all the measurement points. The inclination of this line is the estimated skew, and its offset at $t = 0$ is the estimated clock offset. This amounts to solving the following Linear Program:

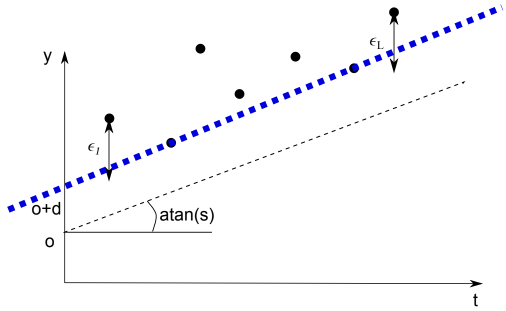

Fig. 1. The one directional Linear Programming algorithm seeks for the line which lies beneath all the measurement points but has the smallest sum of vertical distances to all the points.

Algorithm 1: One Directional LP algorithm [9]

$$\begin{aligned} &\underset{o,s}{\text{minimize}} \quad \sum_{l=1}^{L} \beta(y_l - st_l - o) \\ &\text{subject to} \quad y_l - st_l - o \geq 0, \quad l = 1, \ldots, L \end{aligned} \tag{2}$$

There are L linear constraints stating that all L measurements must be above the estimated line, and the cost function has L terms, summing the distance from the line to the L measurement points. Later, in [10] it was shown that for the one directional measurements case, this algorithm is the Maximum Likelihood estimator. In addition, this algorithm possesses high robustness to exponential noise due to its minimum-like behavior. In [9] its performance was shown to be better than that of Paxson's algorithm presented in [8]. See figure 1 for an illustration of the one directional LP algorithm.

B. Bidirectional Communication

However, in one directional communication it is impossible to separate offset from additional constant network delay, unless the delay is known (or zero). To estimate the offset as well, bidirectional communication must be used. In bidirectional communication, clock C_2 also sends L "incoming" messages back to C_1. Message l is sent at C_2's time ξ_l. Recalling the offset and skew of clock C_2 from the reference time we get $\xi = s\tilde{\tau}_l + o$ where $\tilde{\tau}$ is the reference time at the sending moment. The message reaches clock C_1 after traversing the network and suffering constant propagation delay d. Thus, upon receiving the message clock C_1 shows the time $\tau_l = C_1(\tilde{\tau}_l + d + \eta_l) = \tilde{\tau}_l + d + \eta_l$, the delay d is measured in the receiver's clock. Hence, the relation between the sender and the receiver times is:

$$\xi_l = s\tau_l - d - \eta_l + o \tag{3}$$

where η_l represents the variable portion of the propagation delay and the measurement error, $\eta_l \sim \mathrm{Exp}(\beta^{-1})$.

In [4] it was proposed to solve two independent Linear problems, one for the outgoing messages and one for the incoming messages, and to take the average between the solutions. This results in the following algorithm:

Algorithm 2: Bidirectional LP algorithm (BLP) [4]

$$\underset{o_1,s_1}{\text{minimize}} \quad \sum_{l=1}^{L} \beta(-s_1 t_l - o_1)$$
$$\text{subject to} \quad y_l - s_1 t_l - o_1 \geq 0, \quad l = 1, \ldots, L \tag{4}$$
$$\underset{o_2,s_2}{\text{minimize}} \quad \sum_{l=1}^{L} \beta(s_2 \tau_l + o_2)$$
$$\text{subject to} \quad s_2 \tau_l + o_2 - \xi_l \geq 0, \quad l = 1, \ldots, L \tag{5}$$
$$\text{Output}: \quad o = (o_1 + o_2)/2, \quad s = (s_1 + s_2)/2 \tag{6}$$

This algorithm cannot be shown to be a Maximum Likelihood estimator for this problem but it showed good performance in [4] relative to the authors' implementation of a Kalman filter and relative to simple averaging of the measured delays.

III. Max Margin Algorithms for Clock Synchronization

In this section we present our proposed algorithms for clock synchronization, based on Max Margin optimization.

A. *MM1-LP: Linear Max Margin Algorithm*

The first Max Margin algorithm we present is similar in spirit to the BLP algorithm we presented earlier. Our algorithm too seeks to minimize the vertical difference between the estimated line and the measurement points. The difference is that in the bidirectional LP algorithm two separate lines are estimated, each minimizing the sum of vertical distances between the line and the measurement points. Our algorithm, on the other hand, seeks a single line to begin with, and seeks it so that it has the maximal margin to the closest measurement points of both outgoing and incoming measurements. See figure 2 for an illustration.

Our algorithm takes a very simple form – a Linear Program very similar to the one solved in the bidirectional LP algorithm. The estimated line has to satisfy all the linear constraints, and we demand that it stays at least M away from all the constraints and seek for the maximal M possible. The mathematical formulation is as follows:

Algorithm 3: MM1-LP

$$\underset{o,s}{\text{maximize}} \quad M$$
$$\text{subject to} \quad s\tau_l + o - \xi_l \geq M, \quad l = 1, \ldots, L$$
$$y_l - st_l - o \geq M, \quad l = 1, \ldots, L \tag{7}$$

In the simulations section we will show that this algorithm outperforms state of the art synchronization algorithms.

B. *MM2-QP: Quadratic Max Margin Algorithm*

Instead of using the vertical distance between the lines, we may choose to use the Euclidean distance, as used in SVM. We denote this algorithm by MM2-QP. This algorithm showed performance very similar to that of MM1-LP and thus we leave its details to [16].

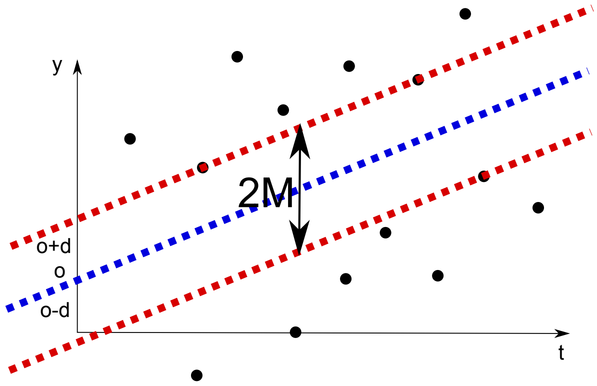

Fig. 2. MM1-LP seeks for two parallel lines with the greatest vertical distance between them that both lie beneath all the outgoing measurements and above all the incoming measurements.

C. *MM3-AP: Approximate Max Margin Algorithm*

The two Max Margin algorithms discussed above provide excellent estimation results, as well as some additional benefits which are discussed below. However, both algorithms need to have the measurements of both clocks to perform their optimization. This is a disadvantage relative to the bidirectional LP algorithm which works independently on each clock's measurements and then only sends the computed offset and skew to the other clock for averaging.

Here we present an approximate algorithm which combines the advantages of both approaches. It is distributed and does not require passing all the measurements of both nodes to a central processor like the bidirectional LP on one hand, but on the other hand it uses Maximum Margin to gain synchronization accuracy.

As we will show in the simulations section, the bidirectional LP and the Max Margin algorithms provide similar performance in skew estimation. It is the offset estimation where the max margin algorithms have significant superiority. Thus our approximated algorithm has two stages:

Algorithm 4: MM3-AP

1) Calculate the skew according to Algorithm 2, i.e. each node calculates a skew using a Linear Program and its own measurements only. The total skew is calculated as the average of the two skew values.
2) Find the offset according to a Max Margin optimization using the calculated skew, see (10) below.

Let us elaborate on step 2 of this algorithm. First, we note that this step is identical to the optimization problem in (7), except that the skew s is obtained from the first step and not optimized. Since s is given, the Maximum Margin parallel lines have a given slope s. Thus, to find the maximum margin they will strive to move away from one another until they meet a single measurement point of the outgoing messages and a single point of the incoming messages respectively. These points are the lowest point among the outgoing messages and the highest point among the incoming messages if we rotate the plane (t, y) by $atan(s)$ clockwise. Recall the measurement model from (1) and (3). The first measurement point of the outgoing messages the Maximum Margin line will meet going up is $\min_{l=1,\ldots,L}(y_l - st_l)$.

Assume this minimal difference was obtained for $l = l'$, then this difference is equal to:

$$\Delta_1 = y_{l'} - st_{l'} = d + \epsilon_{l'} + o \quad (8)$$

Similarly, the first measurement point of the incoming messages the Maximum Margin line will meet going down is $\max_{l=1,\ldots,L}(\xi_l - s\tau_l)$. Assume this maximal difference was obtained for $l = l''$, then this difference is equal to:

$$\Delta_2 = \xi_{l''} - s\tau_{l''} = -d - \eta_{l''} + o \quad (9)$$

Thus, to find the offset by Maximum Margin we simply average the minimal and maximal differences correspondingly achieving:

$$\hat{o}_{\mathrm{MM3-AP}} = (\Delta_1 + \Delta_2)/2 = o + s(\epsilon_{(1)} - \eta_{(1)})/2 \quad (10)$$

where $\epsilon_{(1)}, \eta_{(1)}$ are the minimal values of noise attained in the outgoing and incoming messages correspondingly. This algorithm is fully distributed and very simple. In addition, we will show in the simulations sections it performs almost as well as the exact Max Margin algorithms.

IV. Offset Error Analysis

In this section we analytically derive the MSE for the estimation of the offset by the BLP algorithm and by our Max Margin algorithms for the simple case when the skew is known. Error analysis for the skew estimation of all the above mentioned algorithms appears to be a very difficult task. We therefore provide error analysis for the offset estimation in the case when the skew is given. This analysis becomes explicit in the case of exponential measurement noise, due to the fortunate fact that the minimum of an ensemble of random exponential variables is itself a random exponential variable. In addition, our error analysis is elegant since, as we will show, in the case when the skew is given, several of the above mentioned algorithms estimate the offset in the same manner, and thus our error analysis is compatible for all of them.

As we showed in the development of MM3-AP, when the skew is known, the Maximum Vertical Margin algorithm becomes simply finding the minimal value of $\{y_l - st_l\}_{l=1,\ldots,L}$ and the maximal value of $\{\xi_l - s\tau_l\}_{l=1,\ldots,L}$. It is easy to see that the bidirectional LP algorithm behaves the same with known skew. It seeks to minimize the sum of vertical distances between the measurement points and the estimated lines, while maintaining the constraints saying that the outgoing (incoming) measurements must be above (below) the estimated lines. Thus the estimated lines will again be the highest and lowest possible lines with slope s which touch the lowest $y_l - st_l$ and the highest $\xi_l - s\tau_l$. The offset estimation is then the average of the lines' offsets and the result is again the same as in (10). Likewise, MM2-QP will also have exactly the same estimation, since it looks for the two furthest lines with maximal Euclidean distance between them, while satisfying all the constraints. Since the slope of the lines is equal to s the furthest lines by vertical distance will also be the furthest lines by Euclidean distance.

Thus, we have shown that the BLP algorithm and our three Maximum Margin algorithms MM1-LP, MM2-QP and MM3-AP all estimate the offset in the same way when the skew is given. We now turn to analyze the error of this estimation analytically.

Using equation (10) we get that the offset estimation error is:

$$\tilde{o} = \hat{o} - o = s(\epsilon_{(1)} - \eta_{(1)})/2 \quad (11)$$

that is, the error is $s/2$ times the difference between the minima of two samples of L i.i.d. exponential RV's. The minimum of a sample of L i.i.d. exponential RV's with mean β is also an exponential RV with mean β/L. The difference of two i.i.d. exponential RV's is a Laplacian RV with mean 0 and scale parameter b equal to the Exponential RV's mean, in our case $b = \beta/L$. The multiplication by $s/2$ makes the scale parameter $b = (\beta s)/(2L)$.

Hence the estimation error is a Laplacian RV with mean $\mu = 0$ and scale $b = (\beta s)/(2L)$. Thus, the estimation is unbiased and the MSE is:

$$\mathcal{E}(\tilde{o}^2) = \mathrm{Var}(\tilde{o}) + \mathcal{E}(\tilde{o})^2 = 2b^2 = (\beta^2 s^2)/(2L^2) \quad (12)$$

That means the standard deviation of the estimator is $(\beta s)/(\sqrt{2}L)$. Since $s \sim 1$ we notice the standard deviation is proportional to the mean of the measurement noise and inverse proportional to number of measurements.

V. Robustness to Negative Outliers

The synchronization algorithms discussed above are based on the assumption that the measurement noise values can only be positive. This is due to the fact that the noise is thought to be the excess delay (beyond the constant propagation delay) between the pair of nodes, due to congestion and packet processing overhead. However, in reality a few lower-than-normal delay values might be measured, e.g. due to registration errors or an attack on the network designed to disrupt the synchronization process. The above mentioned algorithms all perform some kind of minimum operation on the measurements, making them extremely vulnerable to negative values of noise. In fact, a single measurement with negative noise would totally change the result of any of these algorithms. In this section we propose a simple modification to our Max Margin algorithm and to the BLP algorithm which can render them robust to negative outliers. Our modification is inspired by the way slacks are added in Support Vector Machines for classification of inseparable classes, see [5].

A. Robust MM1-LP

Similarly to the way slacks are added to the quadratic problem used in SVM, we can add slack variable to the linear program used in MM1-LP. We simply allow every linear constraint to be violated up to a positive slack and add the sum of the slacks to the cost function.

Algorithm 5: Robust MM1-LP

$$\begin{aligned} \underset{o,s}{\text{maximize}} \quad & M - C\left(\sum_{l=1}^{L}\rho_l + \sum_{l=1}^{L}\sigma_l\right) \\ \text{subject to} \quad & s\tau_l + o - \xi_l + \rho_l \geq M, \quad l = 1,\dots,L \\ & y_l - st_l - o + \sigma_l \geq M, \quad l = 1,\dots,L \\ & \rho_l, \sigma_l \geq 0, \quad l = 1,\dots,L \end{aligned} \tag{13}$$

Robust BLP and MM2-QP The bidirectional LP algorithm and MM2-QP may be made robust in a similar way by adding positive slack variables. We leave the details to [16].

VI. Discussion of Basic Properties

Here we would like to discuss existing synchronization algorithms and our own.

a. Robustness to Spiky Noise: The delay measurements between a pair of clocks are prone to very strong noise effects with occasional high valued spikes. The averaging approach taken in [20] is good for two-sided Gaussian noise but gives too much weight to noise spikes. The next idea for dealing with the spiky noise was Paxson's in [7], where local minima and medians are used. This solution is more robust to spiky noise, but each minimum and median operation is performed on a small number of measurements, which may still lead to a high estimation of the minimum. In [9] Moon, Skelly and Towsley improved the robustness further by searching for the line which lies beneath all the measurement points. However, the cost function of the linear programming problem still depends on the values of all the measurements. Finally, our new algorithms use Max Margin optimization. In our algorithms the cost function depends on the width of the band separating the outgoing and incoming measurements only. This means that our algorithms are perfectly robust to spiky noise - only the measurements with the lowest noise values effect the result and determine the line which is the solution to the problem, while the measurements with high values of noise are ignored.

b. Robustness to Synchronization Attacks: Malicious attacks on network synchronization are a serious concern for distributed networks, see [21] for a review. Our algorithms do not provide protection against attacks which change the values of all the measurements. However, if an attack consists of a few exceptionally large or small delay values designed to disrupt the synchronization, then our methods provide a good level of robustness as we have shown earlier.

c. Connection to the Convex Hull: Our Max Margin algorithms may take as input only the points which lie on the lower (upper) boundary of the convex hull of the outgoing (incoming) measurements. This may speed up the algorithms and the network communication required to execute them.

d. Measurement Requirements: Most of the previous work on synchronization assumes bidirectional messages always arrive in pairs. In contrast, our Max Margin algorithms work on an arbitrary batch of measurements.

For further discussion refer to [16].

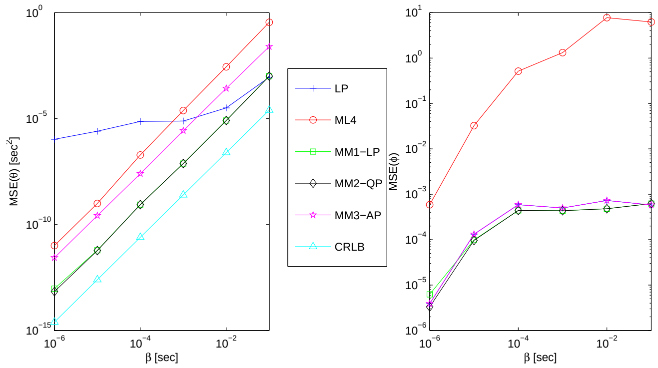

Fig. 3. Offset and skew estimation under different noise levels. MM1-LP and MM2-QP significantly outperform existing algorithms. MM3-AP still has very good performance despite its simplicity. In skew estimation, MM3-AP has the same performance as LP.

VII. Simulation Experiments

To test the performance of our algorithms we compared them to the bidirectional LP algorithm from [4] and to the MLE developed in [15]. We simulate two nodes exchanging messages. The first node is considered as a reference clock, while the second node has offset and skew. The measurements are performed according to the model we have presented in section II. Each experiment is repeated many times to obtain sufficient statistics on the estimators' performance. The mean square error of the offset and skew estimations in all the experiments is plotted for comparison. We plot the performance of the bidirectional LP algorithm ('LP'), the MLE presented in Algorithm 4 in [15] ('ML4'), our Maximum Margin algorithms ('MM1-LP','MM2-QP','MM3-AP') and for the offset estimation plot, also the Cramer Rao lower bound ('CRLB') according to [13] for comparison. In each stage we performed several experiments, changing the mean of the noise while keeping all other parameters constant. The different stages are designed to test different aspects of the algorithms' performance and they are planned as follows:

- Stage 1: Only basic algorithms and no negative outliers.
- Stage 2: Including modified robust algorithms.
  - Stage 2a: With negative outliers.
  - Stage 2b: No negative outliers.

Stage 1 – Only Basic Algorithms and No Negative Outliers: See figure 3 for the results of these experiments.

Stage 2a – With Negative Outliers: In this set of experiments we test the robustness of the algorithms to negative outliers. We assume that most noise values are positive according to the noise model, but several noise values are negative outliers. In our simulations we used 10% outliers, each outlier being a negative exponential variable with the same mean as the measurement noise. Hypothetically, an algorithm may identify these few outliers and exclude them from the estimator's input. We compare our modified robust algorithms (denoted by the prefix 's') against this perfect algorithm (denoted by the prefix 'fk'). The results are presented in figure 4.

Stage 2b – No Negative Outliers: To finalize our simulations, we test our modified algorithms for the case where

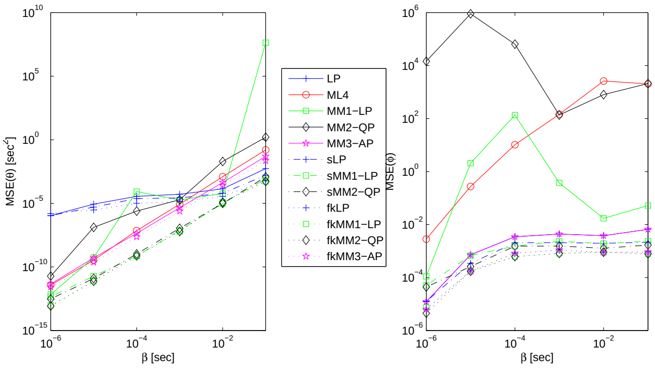

Fig. 4. Offset estimation under different noise levels with negative outliers. The standard unmodified algorithms perform very poorly, while our modified robust algorithms deal well with the outliers, and perform almost as well as algorithms with full knowledge on the outliers' positions.

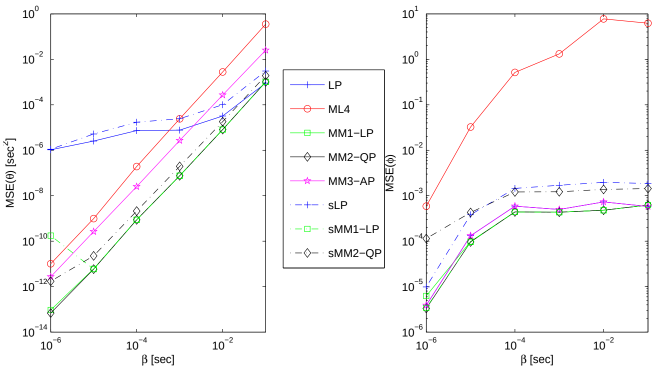

Fig. 5. Offset estimation under different noise levels without negative outliers. The modified algorithms with slacks perform less due to mistaking some of the measurements for outliers. However, the degradation is not severe and might be worth the risk if outliers are expected. Notice the degradation for MM1-LP is negligible

actually no negative outliers exist, to make sure that introducing slacks into the optimization does not cause great damage in the case when no outliers appear in the measurements. See the results in figure 5.

VIII. Conclusion

In this article we have presented novel clock synchronization algorithms based on Maximum Margin. Our Linear algorithm MM1-LP and the related Quadratic algorithm MM2-QP outperform state of the art algorithms, while the third one is an approximation which still has very good performance, but in addition, requires simpler calculations and can be performed in a distributed manner in the individual nodes to be synchronized, with minor exchange of measurement data. We then proposed how to add robustness to our proposed algorithms (as well as to the existing bidirectional LP algorithm) to negative-valued noise outliers without major additions to the optimization problems to be solved. The offset estimation error of the presented algorithms for the special case when the skew is given was derived.

Future Work: An important problem left to be solved is error analysis of the algorithms both for offset and skew estimation. There exists literature on synchronizing networks by exploiting network constraints, see ,e.g., [22]. We are currently working on ways to exploit the same network constraints to improve the performance of existing algorithms such as the Linear Programming algorithm and our Max Margin algorithms presented above. We believe that exploiting these network constraints may give a significant boost in performance, even if we only use small cliques in the network for the constraints data.

References

[1] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," Jul. 2008.

[2] D. Mills, "Network time protocol (version 3) – specification, implementation and analysis, RFC 1305." Technical Report, University of Delaware, 1992.

[3] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[4] A. Bletsas, "Evaluation of kalman filtering for network time keeping," in *Proc. PerCom: 1st IEEE Pervasive Computing and Communication Conference*, pp. 289–296, 2003.

[5] J. Shawe-Taylor and N. Cristianini, *Support Vector Machines and other kernel–based learning methods*. Cambridge University Press, 2000.

[6] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[7] V. Paxson, *Measurements and Analysis of End–to–End Internet Dynamics*. PhD thesis, University of California, Berkeley, 1997.

[8] V. Paxson, "On calibrating measurements of packet transit times," in *Proc. ACM SIGMETRICS'98: Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 11–21, 1998.

[9] S. B. Moon, P. Skelly, and D. F. Towsley, "Estimation and removal of clock skew from network delay measurements," in *IEEE INFOCOM'99: 18th Conference on Computer Communications*, pp. 227–234, 1999.

[10] T. Trump, "Maximum Likelihood trend estimation in exponential noise," *IEEE Trans. Signal Processing*, vol. 49, pp. 2087–2095, Sep. 2001.

[11] H. S. Abdel-Ghaffar, "Analysis of synchronization algorithms with time–out control over networks with exponentially symmetric delays," *IEEE Trans. Communications*, vol. 50, pp. 1652–1661, Oct. 2002.

[12] D. R. Jeske, "On maximum–likelihood estimation of clock offset," *IEEE Trans. Communications*, vol. 53, pp. 53–54, Jan. 2005.

[13] K.-L. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two–way timing message exchanges for wireless sensor networks," *IEEE Trans. Communications*, vol. 55, no. 4, pp. 766–777, 2007.

[14] Q. M. Chaudhari and E. Serpedin, "Clock offset and skew estimation in wireless sensor networks with known deterministic delays and exponential nondeterministic delays," *ICDT'08: International Conf. Digital Telecommunications*, pp. 37–40, 2008.

[15] Q. M. Chaudhari, E. Serpedin, and K. Qaraqe, "On Maximum Likelihood estimation of clock offset and skew in networks with exponential delays," *IEEE Trans. Signal Processing*, vol. 56, 2008.

[16] D. E. Pinkovich and N. Shimkin, "Max margin sync. 2011a." Technical Report, Electrical Engineering Department, Technion, Jan. 2011. Available at: http://webee.technion.ac.il/people/shimkin/PREPRINTS/MaxMarginSync2011a.pdf.

[17] S. Narasimhan and S. S. Kunniyur, "Effect of network parameters on delay in wireless ad–hoc networks." Technical Report, University of Pennsylvania, 2004.

[18] A. Papoulis, *Probability, Random Variables and Stochastic Processes, 3rd Ed.* McGraw–Hill:Columbus, OH, USA, 1991.

[19] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering, 2nd Ed.* Addison–Wesley: Reading, MA, USA, 1993.

[20] J. Elson, L. Girod, and D. Estrin, "Fine–grained network time synchronization using reference broadcasts," in *SIGOPS'02: Operating Systems Review*, vol. 36, pp. 147–163, 2002.

[21] M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in *Proc. SASN 05: The 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2005.

[22] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. 45th IEEE Conf. Decision and Control*, pp. 2734–2739, Dec. 2006.