Implementation and deployment of Kalman Filter - BMTF in 2018

Ioanna Papavergou
on behalf of the BMTF group
National and Kapodistrian University of Athens
Outline

Introduction:
- Motivation
- Algorithm
- Firmware Implementation

Deployment:
- Integration
- Validation
- DQM

Performance
- Standalone muons
- “Special” cases
- Displaced muons

Status and plans - Conclusion
When it all started

- Kalman Filter was already in use offline in CMS and is widely used for track reconstruction in particle physics experiments

- A Kalman Filter algorithm in the L1 Barrel Muon Trigger was proposed by UCLA in *Phase-2 L1 Upgrade Workshop, Madison in June 2017*, for the Phase II upgrade

- UCLA, Uni. of Ioannina and Uni. of Athens, started working toward implementing the Kalman Filter on the L1 BMT already in 2018

- Challenging and very promising plan
• The BMTF has 4 muon stations: in every hit -> position ($\phi$) and bending angle ($\phi_b$)

• Legacy algorithm: momentum assignment through Look Up Tables (LUTs) that use info from 2 stations and vertex constraint
  - vertex constraint improves momentum resolution but is suboptimal for displaced particles

Motivation: Can we do better?
What we want for the HL-LHC

* The DT electronics will be upgraded providing better position and time resolution

* Goal n. 1: Improve resolution by including information from more than 2 stations in the fit
  This cannot be achieved with LUTs - not enough space for all the needed info

* Goal n. 2: Implement momentum assignment without vertex constraint
  This is motivated by physics searches for displaced particles
**Sequential Algorithm**
- Propagate track from station to station (out-in) and match with a stub
- Update track and continue

* After reaching station 1 -> save measurement without vertex constraint
* Propagate to vertex and update -> save vertex constrained measurement
Propagation and update

- Energy losses only considered when propagating to the vertex
- Propagated state is compared to the measurement by the closest stub. Lots of matrix algebra to be done in the firmware

\[
X_n = \begin{pmatrix} k \\ \phi \\ \phi_b \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & b \\ c & 0 & d \end{pmatrix} \begin{pmatrix} k \\ \phi \\ \phi_b \end{pmatrix}_{n-1}
\]

\[
y_n = z = Hx_n
\]

\[
P_{n+1} = FP_{n}F^{T} + Q
\]

\[
Z_{k} = \begin{pmatrix} \phi_s \\ \phi_{bs} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} k \\ \phi \\ \phi_b \end{pmatrix}
\]

\[
S = HPHT + R
\]

\[
K = PHST^{-1}
\]

\[
x = x_n + Kz_n
\]

BUT we do not do the matrix algebra, we look up the Kalman gain matrix.
Kalman gain depends on the curvature for each pattern (shown in next slide).

The new state is the previous state corrected for the Kalman gain times residuals.
A well defined track requires at least two stubs → 11 possible tracks implemented in parallel
The firmware implementation

- Track propagation from station to station (out-in), match with a stub, update the track
- Each track update, in those 11 chains, corresponds to different precalculated Kalman gain mapped in LUTs (reduced amount of matrix operations)
- 3 muon candidates per wedge
- Ghost cleaning → sorting → μGMT

Firmware implemented in the same FPGA of the legacy algorithm (reduced matrix operations allow that)

Algorithms running in parallel for data taking (legacy for triggering) already in Run II
Algorithm Deployment
Integration in CMS

- Firmware integrated in CMS data taking, in parallel with legacy algorithm
- Both algorithms implemented into the same FPGA and take the same data
- Legacy (BMTF) used for triggering and KMTF read-out in DAQ for the collected events

Advantages:

- Real data to study the algorithm planning to deploy it online as default track finder in Run III
- Emulator implemented in CMSSW10X with fixes in versions 10_1_1 - 10_2_1
- Fully updated emulator with all the fixes in the latest version (CMSSW_10_2_1 version 2.71)
Integration in CMS and resource utilisation

- The firmware is implemented using Vivado High Level Synthesis (HLS)
- Vivado compiles C code to HDL
- Uses DSP slices in FPGAs to perform operations minimising resource utilisation

Resource utilisation for both track finders in the FPGA

<table>
<thead>
<tr>
<th>LUT</th>
<th>FF</th>
<th>BRAM</th>
<th>DSP</th>
<th>I/O</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>59%</td>
<td>24%</td>
<td>50%</td>
<td>25%</td>
<td>23%</td>
<td>52%</td>
</tr>
</tbody>
</table>

Both track finders exploit 60% of the FPGA
KF-BMTF resource utilisation

Running both Kalman-Filter and legacy algorithm simultaneously during 2018 data taking on XILINX Virtex 7-690T FPGAs

Kalman latency = 9.25 BXs
Legacy latency = 6.50 BXs
Firmware latency and timing closure

<table>
<thead>
<tr>
<th>@ P5</th>
<th>Triggering algorithm</th>
<th>Kalman IP version</th>
<th>Trigger Latency</th>
<th>Slack</th>
<th>Algorithm CLK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosmics runs</td>
<td>Kalman</td>
<td>Kalman v2.71</td>
<td>9.25 BXs</td>
<td>+ 0.004 ns</td>
<td>160 (MHz)</td>
</tr>
<tr>
<td>p-p runs*</td>
<td>Legacy</td>
<td>Kalman v2.71</td>
<td>6.5 BXs</td>
<td>+ 0.035 ns</td>
<td>160 (MHz)</td>
</tr>
</tbody>
</table>

Minimum achieved Kalman latency but very tight timing closure

* exception Run 325113

Total available BMTF algorithmic latency: 10.5 BXs
Legacy BMTF algorithmic latency:
- 6.5 BXs (contingency 4 BX)
Kalman-MTF algorithmic latency:
- 9.25 BXs (contingency 1.25 BX)
Online deployment

• BMTF unpacker and DQM are updated

  • Based on the triggering firmware the corresponding unpacker and emulator are used in DQM
    - Switching algorithms between triggering and parallel running algorithm without impact on CMS running

• Firmware deployed and tested in cosmic data as default trigger

• Also some p-p data were collected with Kalman algorithm as the default trigger
Algorithm Validation
Collision data 2018

Kalman Muon Trigger as default trigger compared to emulator

\[ p_T \text{ measurement without vertex constraint} \]
Algorithm Validation
Collision data

Collision data 2018

Very good FW-emulator agreement (99.7 %)

Kalman Muon Trigger as default trigger compared to emulator
Unpacker and emulator in DQM

- Unpacker and emulator in DQM
- Shifter can look at KMTF plots

Indicative plots shown the Shift workspace (L1T/L1TStage2BMTF) when the Kalman muon trigger is triggering in cosmic data (these plots appear empty when the BMTF is triggering)
Performance
Standalone muons Efficiency

2017 data

- For efficiency study: SingleMuon with tag and probe method
- K-BMTF calibrated to BMTF (aprox)
- K-BMTF more efficient than BMTF in the gaps \(|0.1| < \eta < |0.4|\)

\(p_T^{L1} > 25 \text{ GeV}\)

\(p_T^{L1} > 30 \text{ GeV}\)
Standalone muons
Resolution

Similar or slightly better resolution than the legacy algorithm
Standalone muons
Rates

Similar or slightly better efficiency and rate @ 20 GeV than the legacy algorithm
Special Cases

High-\(p_T\) efficiency

High efficiency up to high \(p_T\) values for both barrel muon track finders

\(p_T^{L1} > 25\) GeV
Special Cases
Low-\(p_T\) efficiency

- \(J/\psi\) Monte Carlo sample
- Dimuon mass used for efficiency
- \(p_{\text{T}^\text{gen}} > 3\text{GeV}\)
- \(p_{\text{T}^\text{L1}} > 5\text{GeV}\)
- \(J/\psi\) \(p_{\text{T}^\text{gen}} > 8\text{GeV}\)

- Higher efficiency of KF-BMTF in the Gaps
- Overall good efficiency (same or slightly better than BMTF)
Special Cases
Dimuon efficiency

- Equal or better efficiency of KF-BMTF and BMTF to reconstruct low mass dimuon pairs

![Graphs showing dimuon efficiency](image)
Displaced muons
Efficiency in cosmic data

- Cosmic data with collisions menu “bad” proxy for triggering on displaced muons

- Drop in efficiency using vertex constraint at track Dxy > 50 cm
  - Displaced KBMTF regains efficiency at high displacement
  - L1 Dxy = 1 corresponds to 70 cm
  - Kalman filter performs better at higher displacements even with vertex constraint
Displaced muons
Rates in cosmic data

- Vertex unconstrained trigger has higher rate in higher thresholds
  - Resolution 25% compared to 10% with vertex constraint
    - Inner Track veto is needed for lower thresholds
- Very low rates for displaced double muons
  - No track veto needed

Displaced Trigger uses maximum $p_T$ of the constrained and unconstrained measurements
Summary
Status and plans

* Kalman filter algorithm for tracking in the CMS Barrel Muon system has been implemented in the data taking in Run II.
* The algorithm has been studied in detail in CMS data taking.
* Fully commissioned for the CMS Run III.
* KMTF algorithm efficient both in stand-alone and displaced muons.

- Expected benefits:
  * Searches for New Physics with displaced muons.
  * High efficiency - good resolution - low rates.
Backup slides
Single Muon Data set
Tag and Probe selection cuts

- Tag and Probe method
- $dR$ for L1-Reco matching = 0.3
- $dR$ tag-probe = 0.6
- $p_T^{L1}$ cut = 25 GeV
- Reco Iso cut = 0.15
- Reco $p_T$ cut = 30 GeV
- Reco muon $\eta$ and $\phi$ extrapolated to second station