# FPGA Acceleration of Kalman Filter

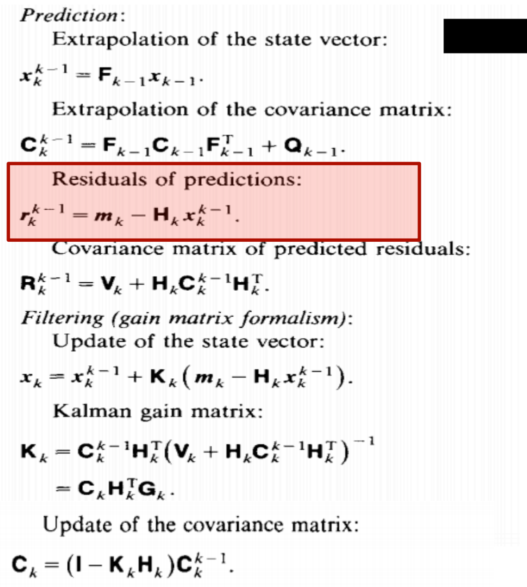
Sioni Summers (Imperial College)

Lukas Arnold (ETH Zurich)

Shahzad Muzaffar (Fermilab)

# Kalman Filter Algorithm

## Formula



*Prediction*:

Extrapolation of the state vector:

$$\boldsymbol{x}_k^{k-1} = \mathbf{F}_{k-1}\boldsymbol{x}_{k-1}.$$

Extrapolation of the covariance matrix:

$$\mathbf{C}_k^{k-1} = \mathbf{F}_{k-1}\mathbf{C}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}.$$

Residuals of predictions:

$$\boldsymbol{r}_k^{k-1} = \boldsymbol{m}_k - \mathbf{H}_k\boldsymbol{x}_k^{k-1}.$$

Covariance matrix of predicted residuals:

$$\mathbf{R}_k^{k-1} = \mathbf{V}_k + \mathbf{H}_k\mathbf{C}_k^{k-1}\mathbf{H}_k^T.$$

*Filtering (gain matrix formalism)*:

Update of the state vector:

$$\boldsymbol{x}_k = \boldsymbol{x}_k^{k-1} + \mathbf{K}_k\left(\boldsymbol{m}_k - \mathbf{H}_k\boldsymbol{x}_k^{k-1}\right).$$

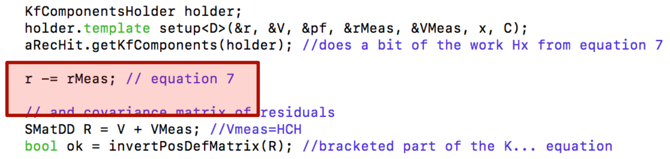
Kalman gain matrix:

$$\mathbf{K}_k = \mathbf{C}_k^{k-1}\mathbf{H}_k^T(\mathbf{V}_k + \mathbf{H}_k\mathbf{C}_k^{k-1}\mathbf{H}_k^T)^{-1} = \mathbf{C}_k\mathbf{H}_k^T\mathbf{G}_k.$$

Update of the covariance matrix:

$$\mathbf{C}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{C}_k^{k-1}.$$

## C++ Code

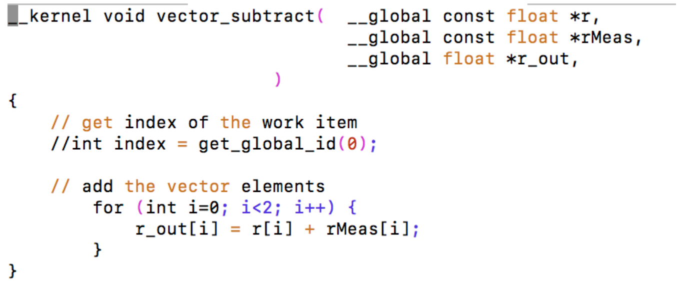


```
KfComponentsHolder holder;
holder.template setup<D>(&r, &V, &pf, &rMeas, &VMeas, x, C);
aRecHit.getKfComponents(holder); //does a bit of the work Hx from equation 7

r -= rMeas; // equation 7

// and covariance matrix of residuals
SMatDD R = V + VMeas; //Vmeas=HCH
bool ok = invertPosDefMatrix(R); //bracketed part of the K... equation
```

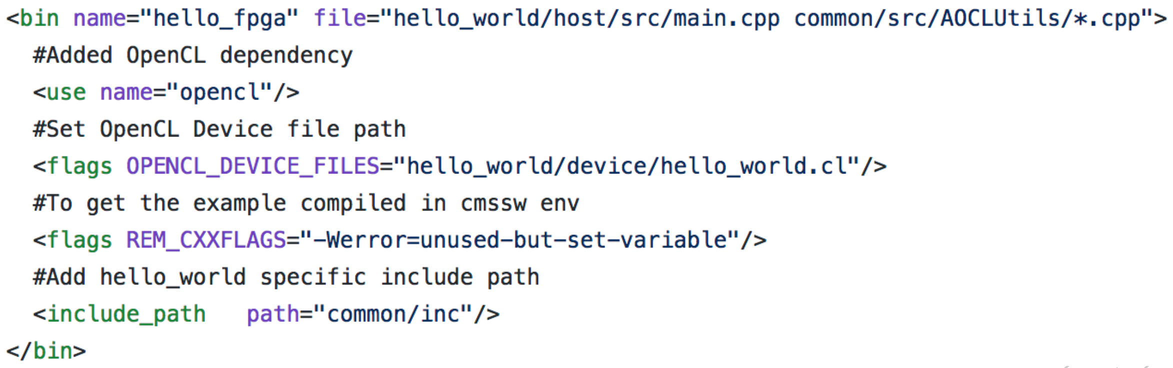
## OpenCL Code



```
__kernel void vector_subtract(  __global const float *r,
                                __global const float *rMeas,
                                __global float *r_out,
                             )
{
    // get index of the work item
    //int index = get_global_id(0);

    // add the vector elements
        for (int i=0; i<2; i++) {
            r_out[i] = r[i] + rMeas[i];
        }
}
```

## Integration to SCRAM



```
<bin name="hello_fpga" file="hello_world/host/src/main.cpp common/src/AOCLUtils/*.cpp">
  #Added OpenCL dependency
  <use name="opencl"/>
  #Set OpenCL Device file path
  <flags OPENCL_DEVICE_FILES="hello_world/device/hello_world.cl"/>
  #To get the example compiled in cmssw env
  <flags REM_CXXFLAGS="-Werror=unused-but-set-variable"/>
  #Add hello_world specific include path
  <include_path    path="common/inc"/>
</bin>
```

# Next steps

- Start using update SCRAM version
- Write host code