```
(defquery kalman
    "A basic Kalman smoother. Predicts a state sequence from the posterior
    given observations"
    [observations obs-matrix obs-cov trans-matrix trans-cov init-mean init-cov]
    (let [;; D is dimensionality of data,
          ;; K is dimensionality of latent space
          [D K] (shape obs-matrix)
          ;; prior on observation noise
          obs-dist (mvn (zero-vector D) obs-cov)
          ;; prior on initial state
          start-dist (mvn init-mean init-cov)
          ;; prior on transition noise
          trans-dist (mvn (zero-vector K) trans-cov)]
    (predict :states
      (matrix
        (reduce (fn [states obs]
                  (let [;; sample next state
                        prev-state (peek states)
                        state (if prev-state
                                (add (mmul trans-matrix prev-state)
                                     (sample trans-dist))
                                (sample start-dist))]
                    ;; observe next data point (when available)
                    (observe (count states) obs-dist (sub (mmul obs-matrix state) obs))
                    ;; append state to sequence and continue with next obs
                    (conj states state)))
                ;; start with empty sequence
                []
                ;; loop over data
                observations)))))
```