

«Фильтр Калмана для „чайников“: матрица наблюдаемости и фундаментальная матрица»

Студентам каких-либо радиотехнических специальностей часто читают и даже пытаются объяснить такой алгоритм фильтрации, как рекуррентный байесовский фильтр, типа фильтр Калмана. Сам знаю, что на первых порах такие алгоритмы крайне тяжело воспринимаются. И как правило сей материал студентам преподносят на матером техническом языке. И пока ты сидишь и вспоминаешь значение того или иного слова, понимание сути этого алгоритма до тебя так и не доходит. Единственное, что в итоге запоминается, что этот фильтр придуманный Калманом очень хорошая штука, которая почти нигде не используется, потому что он трудно реализуем в аппаратуре.

Действительно, реализация фильтра Калмана на борту космического аппарата или в навигационной аппаратуре потребителя была затруднительной, так как требовался достаточно мощный процессор и приличный объем памяти, т.е. значительное увеличение габаритов аппаратуры и, соответственно, массы.

Как правило, использовался наиболее простой (и в реализации) метод наименьших квадратов (МНК), алгоритм фильтрации по полной выборке.

По сравнению с фильтром Калмана, МНК не использует априорной информации о движении объекта и оценка состояния выполняется на основе выборки измерений. Т.е. до получения первой качественной оценки состояния объекта помимо времени на получение информации от навигационных спутников (альманахи, эфемериды и т.д.) прибавляется время на накопление измерений. Навигаторы с МНК плохо определяют именно движущиеся объекты, но если, с уже включенным приемником, вы постоите минут 5 на одном месте, тогда навигатор может показать ваше местоположение с точностью до сантиметров.

Фильтр Калмана, в свою очередь, после получения первых измерений, в состоянии достаточно качественно сделать оценку движение объекта. Априорная информация о движении объекта в алгоритме выражается в матрицах наблюдаемости (H) и фундаментальной матрице (Φ).

Итак. Вот, собственно, сам фильтра Калмана (точнее, одно из его представлений):

$$q_{i+1}^* = \hat{q}_{i+1} + K_{i+1} [R_{H(i+1)} - R_{0(i+1)}],$$

$$K_{q(i+1)} = [I - K_{i+1} H_{i+1}] K_{q0(i+1)} [I - K_{i+1} H_{i+1}]^T + K_{i+1} W_{i+1} K_{i+1}^T,$$

$$K_{i+1} = K_{q0(i+1)} H_{i+1}^T [H_{i+1} K_{q0(i+1)} H_{i+1}^T + W_{i+1}]^{-1},$$

$$K_{q0(i+1)} = \Phi_i K_{qi} \Phi_i^T + V_i.$$

q_{i+1}^* - апостериорный вектор состояния, размерностью $b \times 1$;

\hat{q}_{i+1} - априорный вектор состояния, размерностью $b \times 1$;

K_{i+1} - матрица коэффициентов усиления, размерностью $b \times n$;

$R_{H(i+1)}$ - “истинный” вектор измерений, размерностью $p \times 1$;

$R_{0(i+1)}$ - навигационный (расчетный) вектор измерений, размерностью $p \times 1$;
 $K_{q(i+1)}$ - априорная матрица ковариаций среднеквадратических отклонений, характеризующих точность оценки вектора состояния, размерностью 6×6 ;
 H_{i+1} - матрица наблюдаемости, размерностью $p \times 6$;
 I - единичная матрица, размерностью 6×6 ;
 W_{i+1} - матрица ошибок измерения, размерностью $p \times p$;
 $K_{q0(i+1)}$ - апостериорная матрица ковариаций среднеквадратических отклонений, характеризующих точность оценки вектора состояния, размерностью 6×6 ;
 Φ_i - фундаментальная матрица, размерностью 6×6 ;
 V_i - матрица влияния внешних возмущений на вектор состояния.
 n – количество спутников, с которых обрабатываются измерения.

Для простоты рассмотрим фильтр для вектора состояния q из шести элементов (именно поэтому размерности вышеописанных элементов включают в себя цифру «6»): три проекции радиус-вектора (радиус-вектор с началом в центре Земли, направленный в центр масс объекта) и три проекции вектора скорости объекта.

Для студентов, изучающих фильтр Калмана, трудности доставляют, в основном, матрица наблюдаемости и фундаментальная матрица.

В умных книгах иногда такие вещи описываются очень сложно, для понимания простого студента. Ну и с учетом того, что некоторые дисциплины (вроде диф.ур.) слегка позабылись, понимание таких сложных вещей, как «частные производные измеренных параметров в окрестности навигационной траектории» =) становится близко к полному непониманию.

Попробуем объяснить построение этих матриц как можно проще (в пределах разумного).

Матрица наблюдаемости.

Предположим, что мы находимся на борту космического аппарата. Помимо нас на борту есть такие навигационные средства, как ИНС (инерциальная навигационная система, пусть будет платформенная) и спутниковая навигационная система (какой-нибудь GPS/ГЛОНАСС приемник). Предположим, что в каждый момент времени нам с помощью ИНС, известны все параметры нашего движения $q = \{R_x, R_y, R_z, V_x, V_y, V_z, \}$. И вроде бы всё замечательно. Но интуиция, а может эхо давно забытых лекций, подсказывает нам, что эти платформенные ИНС имеют обычай «уходить», т.е. накапливать ошибку. И, значит, со временем станет не очень понятно, как мы вообще движемся. Чтобы скорректировать ИНС к нам на помощь приходит достаточно точный GPS/ГЛОНАСС приемник. (Правда слегка недоделанный: Нам еще самим нужно будет высчитывать псевдодальности и псевдоскорости.) Но и приемник не может выдавать абсолютно точные оценки. А если вспомнить, что наш космический аппарат находится в агрессивной среде, где на него воздействуют нецентральные гравитационные силы Земли, Луны и Солнца.. Может быть там еще есть и верхняя атмосфера Земли. Короче говоря, сплошные ошибки и погрешности. Но чтобы отфильтровать эти ошибки нам как раз пригодится фильтр Калмана. Но вот беда: в алгоритме фильтрации нужно реализовать некую матрицу

наблюдаемости (H), которая связывает измерения GPS/ГЛОНАСС приемника и вектор состояния, получаемый с помощью ИНС. В, общем-то, это некая матрица перехода связывающая наши реальные(по ИНС) местоположение и скорость с измерениями GPS/ГЛОНАСС приемника.

Для тех, кто в танке!

В математике есть такая функция $y=A*x$, так вот, если сопоставить эту функцию с тем, что написано выше, получится: [вектор состояния(навигационный (по ИНС))] = A* [измерения приемника].

Так вот A – это и есть H, матрица наблюдаемости.

Размерность этой матрицы $n \times b$. b - это число элементов вектора состояния q, n – число измерений в конкретный момент времени. Предположим, что наш GPS/ГЛОНАСС приемник видит в данный момент только 3 навигационных спутника. Тогда от этих спутников мы можем получить максимум 6 измерений одновременно: псевдодальность и псевдоскорость от одного спутника, псевдодальность и псевдоскорость от другого и т.д. Следовательно, размерность матрицы в случае 3-х видимых навигационных спутника - 6×6 .

Нужно теперь определить эту матрицу. Роясь в карманах, находим маленькую книжечку. Нам её, незаметно, засунули в центре управления полетами. В ней находим главу «Модель измерений». Как правило, такая модель описывается формулой(в данном случае двумя: для псевдодальности и псевдоскорости). Из чего эти формулы состоят, разбирать не будем. Предположим, что эти формулы есть, и все их компоненты нам понятны.

Заполним первую строку матрицы наблюдаемости. Если перевести текст по заполнению матрицы, из умных книг, и адаптировать его для среднего студента получится приблизительно следующее: возьмем формулу модели измерения для псевдодальности и продифференцируем сначала по R_x - таким образом получим первый элемент первой строки матрицы наблюдаемости, потом продифференцируем по R_y – получим второй элемент матрицы наблюдаемости, первой строки и т.д. до V_z .

Вот как дифференцировать – рассказывать не буду.

Для заполнения второй строки необходимо взять уравнение модели измерения для псевдоскорости и проделать с ней такие же операции.

Таким образом, вперемежку, аналитически заполнится вся матрица наблюдаемости.

Останется только запрограммировать её.

Правильно ли вы составили матрицу наблюдаемости? Её строки, по сути, будут единичными векторами, направленными на навигационный спутник. Так что, если модуль строки матрицы наблюдаемости равен 1, то скорее всего вы правильно продифференцировали.

Фундаментальная матрица

Эта матрица, описывает динамику состояния объекта. На пальцах её суть можно объяснить следующим образом:

Ваш вестибулярный аппарат задаёт связанную с вами, систему координат.

Вот вы стоите неподвижно в углу. Угол комнаты задает инерциальную систему координат. В какой-то момент вы решили развернуться на 180 градусов и делаете это(долго разворачиваетесь, секунд 5). Так вот фундаментальная матрица - это некое описание вашего движения, относительно угла комнаты, с момента времени начала вашего поворота до его окончания, т.е. движение связанной системы координат (в проекциях на оси инерциальной) относительно инерциальной в заданном интервале времени.

Более менее разобрались, что это за матрица такая. Теперь её нужно заполнить...

Предположим, что в сумке, потрепанной студенческой жизнью и пивом, завалялась дискета, которую вы так и не отдали своему согруппнику. На этой дискете к счастью оказался программно реализованный численный метод интегрирования (пусть будет Рунге-Кутты). Вот мы его как раз и применим для формирования фундаментальной матрицы.

Заполнение фундаментальной матрицы.

1. Берется “истинный” вектор состояния на момент времени начала вашего поворота и численно интегрируется на интервале времени в 5 секунд (интегрируется формула, выражающая модель нашего движения).

Полученный вектор состояния запоминается. Назовем его X .

2. Берется “истинный” вектор состояния до вашего поворота и к первой компоненте вектора прибавляется некоторое приращение (к примеру, 100 метров). Пример: $Rx+100$ метров (если Rx в метрах). Остальные компоненты вектора не трогаем.

Полученный вектор состояния интегрируется на 5 секунд. В результате получаем вектор X^* .

3. Покомпонентно из вектора состояния X вычитаем X^* и делим на приращение (также покомпонентно). В результате получаем 1-й столбец фундаментальной матрицы.

4. Для получения остальных столбцов фундаментальной матрицы выполняются 2,3 пункты. (Номер элемента вектора состояния, к которому осуществляется приращение, соответствует номеру столбца фундаментальной матрицы).

Если модель движения выражается формулой для кругового движения и ваш космический аппарат представлен материальной точкой, тогда очевидно на протяжении всего вашего полета, пересчитывая фундаментальную матрицу её значения будут практически одинаковыми.

Вот и всё.