# MOTION ESTIMATION OF OBJECTS IN KC135 MICROGRAVITY

Lisa Hewgill
Lockheed Engineering and Sciences Company
Houston, Texas

## Abstract

The ability of a autonomous space robot to grasp a freely translating and rotating object is being tested in the simulated microgravity environment aboard a KC135 airplane. The Extravehicular Activity Helper/Retriever's (EVAHR's) arm trajectory planner continually requires a current estimate of the target's translational and rotational state. The target's attitude, angular velocity and angular acceleration define its rotational state and the target's translational state include its position, velocity and acceleration. Estimators have been developed based on the extended Kalman filter (EKF) algorithm. The KC135 microgravity environment does not have a convenient inertial reference frame for the translational dynamics and therefore, the translational as well as the rotational object dynamics are described by nonlinear equations. The estimator algorithms require intensive mathematical computation and therefore, I860 microprocessors are used so that the software will run in real time. Estimator design, implementation concerns and issues specific to the KC135 environment are discussed. Translational state estimator performance results from simulation testing and from real-time integrated system testing are presented.

## KC135 Experiment

One of the objectives of the Automation and Robotics Department at the Johnson Space Center is to design and develop a free-flying autonomous space robot. The immediate goal of the EVAHR project is to have the EVAHR grasp a freely translating and rotating object in a microgravity environment. This environment is simulated in the cabin of a KC135 airplane by having the plane fly along a parabolic trajectory. Sections of the robot that are necessary for the experiment are bolted to the KC135 cabin floor or are secured in some other fashion. Included are a Robotics Research arm, an inertial measurement unit (IMU), a vision system, release mechanism, cages containing the I860 processors and other computer equipment. There are two candidate vision systems: PRISM3 (1) and the Perceptron scanner (2). The PRISM3 position measurement rate is 20-30 Hz and the position measurement rate of Perceptron scanner is 5 - 8 Hz.

## KC135 Environment

The gravitational forces experienced in the cabin of the KC135 change from above 1.5 g (1 g = 32.2 ft/s$^2$) during "pull-up" to less than 50 mg of microgravity in approximately 7 seconds. The microgravity environment (less than 100 mg) lasts approximately 20 seconds; however, the microgravity period during which the target is relatively stationary in the Robotics Research arm's work space usually ranges from 0 to 10 seconds per parabola. This decrease is caused by the fluctuations of up to 100 mg in the gravitational acceleration during the microgravity portion of the parabola. These undesired microgravity fluctuations cause the relatively stationary floating target to fly against the ceiling or floor of the airplane.

Another characteristic of the KC135 environment is the usual initial fluctuation in the vertical acceleration as the airplane enters the microgravity portion of the flight. This is illustrated in Fig. 1. To avoid premature target release and the resulting undesired target dynamics, a release indicator mechanism was developed. The vertical acceleration, the pitch rate, and the pitch acceleration are monitored to determine the proper time to release the target. These measurements are taken from accelerometers and 3-axis gyroscopes. The microgravity portions of the flight that experience minimal pitch acceleration tend to be the better quality parabolas.

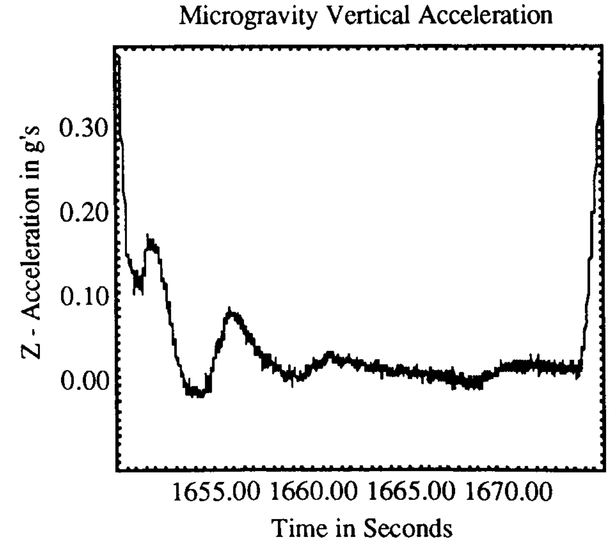


Fig. 1

## Relative Translational Dynamics

The KC135 cabin environment and the EVAHR instrumentation do not provide a practical inertial reference frame which can be utilized in modelling the target's translational dynamics. The EVAHR translational estimator coordinate frame, which is a Cartesian coordinate system centered at a corner of the IMU plate, is essentially an accelerating reference frame. At present, the EVAHR hardware does not include a sensor that gives position expressed in an earth reference frame (differential GPS etc.). Double integrating the output of the accelerometers for 1 - 2 hours without another correcting measurement is undesirable. EVAHR instrumentation does provide enough information to calculate the target 's acceleration relative to the robot. There are several implicit assumptions in determining the relative acceleration. One is that the EVAHR's and target's gravitational accelerations are equal in magnitude. It is also assumed that the only force acting on target is gravity and third, the atmospheric drag acting on the target is zero. The relative acceleration of the target with respect to the EVAHR is given by

$$^{E}\mathbf{a}_T = -\,^{E}\omega_E \times \,^{E}\omega_E \times \,^{E}\mathbf{r}_T - \,^{E}\alpha_E \times \,^{E}\mathbf{r}_T - 2 * \,^{E}\omega_E \times \,^{E}\mathbf{v}_T - \,^{E}\mathbf{a}_E \quad (1)$$

where

$^{E}\omega_E$ is the EVAHR's angular velocity (filtered gyro reading)

$^{E}\alpha_E$ is the EVAHR's angular acceleration

$^{E}\mathbf{a}_E$ is the EVAHR's translational acceleration (filtered accelerometer reading)

$^{E}\mathbf{r}_T$ is the target's position relative to EVAHR

$^{E}\mathbf{v}_T$ is the target's velocity relative to EVAHR

$^{E}\mathbf{a}_T$ is the target's acceleration relative to EVAHR.

For computational reasons, during each calculation time period (iteration) the acceleration is determined twice. The first calculation uses the target's position and velocity estimates from the previous iteration. Next, the position and velocity estimates are updated using the new relative acceleration. The acceleration is then again determined using the updated position and velocity estimates. The relative velocity and relative position are determined in each iteration as follows

(i) compute the relative acceleration, $^{E}\mathbf{a}_T$, based on the target's velocity and position estimates of the previous iteration

(ii) compute

$$^{E}\mathbf{v}_T = \,^{E}\mathbf{vpast}_T + \frac{\Delta t}{2}(^{E}\mathbf{a}_T + \,^{E}\mathbf{apast}_T) \quad (2)$$

$$^{E}\mathbf{r}_T = \,^{E}\mathbf{rpast}_T + \frac{\Delta t}{2}(^{E}\mathbf{v}_T + \,^{E}\mathbf{vpast}_T) \quad (3)$$

where

$^{E}\mathbf{apast}_T$ is the target's relative acceleration during the last iteration

$^{E}\mathbf{vpast}_T$ is the target's relative velocity during the last iteration

$^{E}\mathbf{rpast}_T$ is the target's relative position during the last iteration

$\Delta t$ is length of iteration interval.

(iii) recompute $^{E}\mathbf{a}_T$ based on updated $^{E}\mathbf{v}_T$ and $^{E}\mathbf{r}_T$

(iv) recompute (ii) based on updated $^{E}\mathbf{a}_T$.

## KC135 Translational EKF Filter Design

The EKF system model for the KC135 translational state estimator may be expressed as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t)) + \mathbf{w}(t) \quad (4)$$

and the measurement model may be expressed as

$$\mathbf{z}_k = \mathbf{H}\,\mathbf{x}(t_k) + \mathbf{v}_k \quad (5)$$

where

$\mathbf{x}(t)$ is the state vector

$\mathbf{w}(t)$ is a zero mean white process

$\mathbf{v}_k$ is a zero mean white sequence.

For the EKF algorithm, the state may be written as

$$\mathbf{x}(t) = \mathbf{x}^*(t) + \Delta\mathbf{x}(t) \quad (6)$$

where

$\mathbf{x}^*(t)$ is an estimate of the state.

Combining equations (4) and (6), and expanding to the first order,

$$\frac{d\mathbf{x}^*(t)}{dt} + \frac{d\Delta\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}^*(t)) + \frac{\partial \mathbf{f}(\mathbf{x}(t))}{\partial \mathbf{x}}\bigg|_{\mathbf{x}(t)=\mathbf{x}^*(t)} \Delta\mathbf{x}(t) + \mathbf{w}(t) \quad (7)$$

Therefore, the linearized model maybe expressed as

$$\frac{d\Delta\mathbf{x}(t)}{dt} = \frac{\partial \mathbf{f}(\mathbf{x}(t))}{\partial \mathbf{x}}\bigg|_{\mathbf{x}(t)=\mathbf{x}^*(t)} \Delta\mathbf{x}(t) + \mathbf{w}(t) \quad (8)$$

The state matrix calculation is determined by

$$\Phi(t_{k+1}, t_k) = \mathbf{I} + \mathbf{F}\Delta t + \left(\frac{\mathbf{F}}{dt} + \mathbf{F}^2\right)\frac{\Delta t^2}{2} \tag{9}$$

where

$$\mathbf{F} = \frac{\partial f(x(t))}{\partial x}\Big|_{x(t_k) = x^*(t_k)}$$

$\Delta t$ is the time interval between $t_k$ and $t_{k+1}$

This calculation is extended out to the second order. Accuracy requirements had to be balanced against computer computation time requirements in this determination. The Kalman filter process noise covariance matrix, $\mathbf{Q}(t_{k+1})$, is computed according to

$$\mathbf{Q}(t_{k+1}) = \Phi(t_{k+1}, t_k)\, \mathbf{H}\, \sigma\, \mathbf{H}^T \Phi^T(t_{k+1}, t_k) + \mathbf{Q}(t_k) \tag{10}$$

where

$\sigma$ is a sparse matrix whose nonzero elements are associated with the variances of the white Gaussian noise of equation (20).

The Kalman filter error covariance matrix is propagated according to

$$\mathbf{P}(t_{k+1}) = \Phi(t_{k+1}, t_k)\, \mathbf{P}(t_k)\, \Phi^T(t_{k+1}, t_k) + \mathbf{Q}(t_{k+1}) \tag{11}$$

The state, $\mathbf{x}(t_k+)$, is revised by the filter according to

$$\mathbf{x}(t_k+) = \mathbf{x}(t_k-) + \mathbf{K}(t_k)\,[\mathbf{z}(t_k) - \mathbf{H}\mathbf{x}(t_k-)] \tag{12}$$

and the Kalman gain is given by

$$\mathbf{K}(t_k) = \mathbf{P}(t_k-)\mathbf{H}^T[\mathbf{H}\mathbf{P}(t_k-)\mathbf{H}^T + \mathbf{R}]^{-1} \tag{13}$$

where

$\mathbf{R}$ is the position measurement error covariance matrix.

Also, the Kalman filter error covariance is updated by

$$\mathbf{P}(t_k+) = [\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}]\mathbf{P}(t_k-) \tag{14}$$

The state vector specific to the relative translational state estimator is defined as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \text{ - 3 target position components} \\ \mathbf{v} \text{ - 3 target velocity components} \\ \omega_e \text{ -3 EVAHR gyro errors} \\ \alpha_e \text{- 3 EVAHR angular acc. errors} \\ \mathbf{a}_{2e} \text{- 3 EVAHR accelerometer errors} \end{bmatrix} \tag{15}$$

The IMU readings and the target's position and velocity estimates, all of which have some error, must be used to determine the target's acceleration, which is given by equation (1). Now the measured angular velocity and the EVAHR angular acceleration can be written as

$$\omega = \omega_t + \delta\omega \tag{16}$$

$$\alpha = \alpha_t + \delta\alpha \tag{17}$$

where

$\omega$ is the vector of filtered EVAHR gyro readings

$\omega_t$ is the vector of true EVAHR angular velocities

$\delta\omega$ is the gyro error vector.

$\alpha$ is the vector of computed EVAHR angular accelerations

$\alpha_t$ is the vector of true EVAHR angular accelerations

$\delta\alpha$ is the angular acceleration error vector.

Similarly, the EVAHR acceleration may be written as

$$\mathbf{a}_2 = \mathbf{a}_{2t} + \delta\mathbf{a}_2 \tag{18}$$

where

$\mathbf{a}_2$ is the vector of filtered EVAHR accelerometer readings

$\mathbf{a}_{2t}$ is the vector of true EVAHR accelerations

$\delta\mathbf{a}_2$ is the accelerometer error vector.

To form the state matrix, the time derivative of the target's velocity error is expressed as

$$\frac{d\delta v}{dt} = \mathbf{A}\delta\mathbf{p} + \mathbf{B}\delta\mathbf{v} + \mathbf{C}\delta\omega + \mathbf{D}\delta\alpha - \delta\mathbf{a}_2 \tag{19}$$

where

$$\mathbf{A} = \begin{bmatrix} \omega_1^2+\omega_2^2 & -\omega_1\omega_0+\alpha_2 & -\omega_2\omega_0-\alpha_1 \\ -\omega_0\omega_1-\alpha_2 & \omega_2^2+\omega_0^2 & -\omega_2\omega_1+\alpha_2 \\ -\omega_0\omega_2+\alpha_1 & -\omega_1\omega_2-\alpha_0 & \omega_0^2+\omega_1^2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 2\omega_2 & -2\omega_1 \\ -2\omega_2 & 0 & 2\omega_0 \\ 2\omega_1 & -2\omega_0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

$a_{11} = -\omega_1 p_1 - \omega_2 p_2$

$a_{21} = -\omega_0 p_2 + 2\omega_1 p_0 - 2v_2$

$a_{31} = 2\omega_2 p_0 - \omega_0 p_2 + 2v_1$

$a_{12} = 2\omega_0 p_1 - \omega_1 p_0 + 2v_2$

$a_{22} = -\omega_2 p_2 - \omega_0 p_0$

$a_{32} = -\omega_1 p_2 + 2\omega_2 p_1 - 2v_0$

$a_{13} = 2\omega_0 p_2 - \omega_2 p_0 - 2v_1$

$a_{23} = 2\omega_1 p_2 - \omega_2 p_1 + 2v_0$

$a_{33} = -\omega_0 p_0 - \omega_1 p_1$

$$D = \begin{bmatrix} 0 & -p_2 & p_1 \\ p_2 & 0 & -p_0 \\ -p_1 & p_0 & 0 \end{bmatrix}$$

Therefore, the particular EKF system model is given by

$$\frac{d}{dt}\begin{bmatrix} \delta \mathbf{p} \\ \delta \mathbf{v} \\ \delta \omega \\ \delta \alpha \\ \delta \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\beta_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\beta_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\beta_2 \end{bmatrix} \begin{bmatrix} \delta \mathbf{p} \\ \delta \mathbf{v} \\ \delta \omega \\ \delta \alpha \\ \delta \mathbf{a}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{n}_1 \\ \mathbf{n}_1 \\ \mathbf{n}_2 \end{bmatrix} \tag{20}$$

where

**p** is the estimate of the target's position relative to the EVAHR

$\delta p = p_{true} - p$

**v** is the estimate of the target's velocity relative to the EVAHR

$\delta v = v_{true} - v$

$\beta_1$ a 3x3 diagonal matrix with the value $\beta_1$

$\beta_2$ is a 3x3 diagonal matrix with the value $\beta_2$

$\mathbf{n}_1$ is the white Gaussian noise vector whose value is determined from gyro noise tests and from system testing

$\mathbf{n}_2$ is the white Gaussian noise vector whose value is determined from accelerometer noise tests and from system testing

Equation (20) has the same form as equation (8).

## Rotational State Estimator

The rotational state estimator uses quaternions to describe the target's attitude and utilizes an EKF algorithm [3]. For further information on quaternion estimation refer to Bar-Itzhack's work [4]. Bierman's U-D factorization [5] technique was implemented to test for improved performance. An initial difference in performance was noted between the conventional EKF and the U-D factorization algorithm. However, after convergence, there was not a marked difference in performance.

The essential difference between the rotational state estimator intended for the earth orbit scenario and the KC135 rotational state estimator is the definition of the inertial reference frame. For the KC135, the inertial reference frame for the target's attitude will be the EVAHR's coordinate frame when the microgravity portion of each parabola commences (t=0). The EVAHR's attitude will be determined by integrating the output of the gyros from the commencement time (t=0).

## Results

The performance results shown in Fig. 2 are the product of a simulation test. A program was developed that simulates the dynamics of KC135 airplane, the IMU sensor readings, and the dynamics of the target. The added noise is white Gaussian noise. The Euclidian norm error of the position vector estimate and of the position vector measurement are shown.

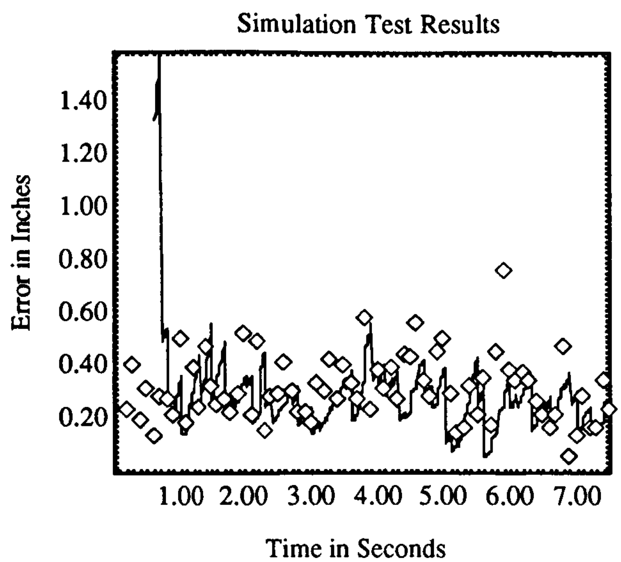


Fig. 2

Testing of the Perceptron scanner is ongoing and calibration of PRISM3 vision system is in process at the writing of this paper. Results shown in Fig. 3 are preliminary. For this test the target was stationary and Fig. 3 shows the target's x-position estimate as well as the x-position measurement. Once calibration is complete, the values of measurement error covariance matrix **R** of equation (14) will be modified.

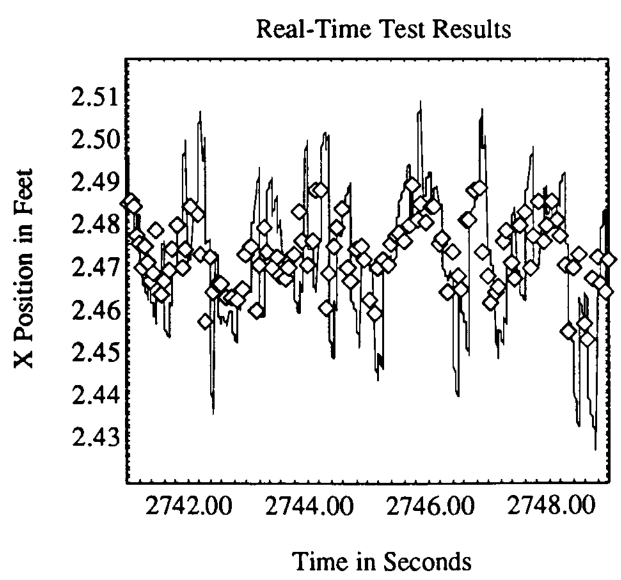


Fig. 3

## Implementation Issues

For the arm trajectory planner software to be able to receive the target's state at the rate of 100 Hz, the relative translational state estimator had to be divided into a repropagation unit and real-time propagation unit. The repropagation unit accepts dated position measurement inputs, compares the measurement to the appropriate archived state, revises the state, and then repropagates the state to present time. The updated state is passed to the real-time propagation unit. This module replaces its target state and Kalman filter parameters with the new updated state and new Kalman filter parameters. The real-time propagation unit continues to propagate the state and appropriate Kalman filter parameters while the repropagation unit has accepted another dated position measurement and is revising another archived estimate. The architecture is shown below in Fig. 4.

The estimators are implemented on Mercury I860 processors because of their fast computation capability. Closed form solutions were used where possible and matrix computation decreased by taking advantage of sparse matrices.

A sensor bias test is performed just prior to the start of the flight, and these values are used to correct the IMU readings. The expected IMU drift during the flight was determined by running drift tests for a length of time comparable to flight duration, (1 - 2 hr). The sensor outputs are also passed through hardware and software filters before the readings are fed into the estimator at the rate of 100 Hz.

## Summary

The design and performance of the KC135 translational state estimator were discussed. Also presented were issues specific to the KC135 microgravity environment.

## Acknowledgements

The author wishes to acknowledge the significant contributions of Mr. Greg Anderson with the development of the KC135 dynamics and of Mr. Robert Norsworthy with the implementation of the translational estimator on the I860 processors, both of whom are colleagues on the EVAHR project.

## References

1. E. Huber, "Object Tracking with Stereo Vision", to be presented at the AIAA Conference on Intelligent Robots for Factory, Field, Service, and Space, Houston, Tx, Mar. 20 - 24, 1994

2. M. Littlefield, "Real-Time Tracking of Objects in a KC-135 Microgravity Experiment", to be presented at the AIAA Conference on Intelligent Robots for Factory, Field, Service, and Space, Houston, Tx, Mar. 20 - 24, 1994

3. L. Hewgill, "Motion estimation of a freely rotating body in earth orbit", SPIE Cooperative Intelligent Robotics in Space III Proceedings, Boston, MA., Nov. 16 -18, 1992. Vol. 1829, pp. 444 - 457

4. I. Y. Bar-Itzhack and Y. Oshman, "Attitude Determination from Vector Observation: Quaternion Estimation", *IEEE Transactions on Aerospace and Electronic Systems*, pp. 128 - 136, Jan. 1985

5. G. H. Bierman, "Measurement updating using the U-D factorization", IEEE Conference on Decision and Control, 6th and Symposium on Adaptive Process, 14th Proceedings, Houston, TX, Dec. 10 - 12, 1975. pp. 337 - 346, Institute of Electrical and Electronic Engineers, New York, 1975

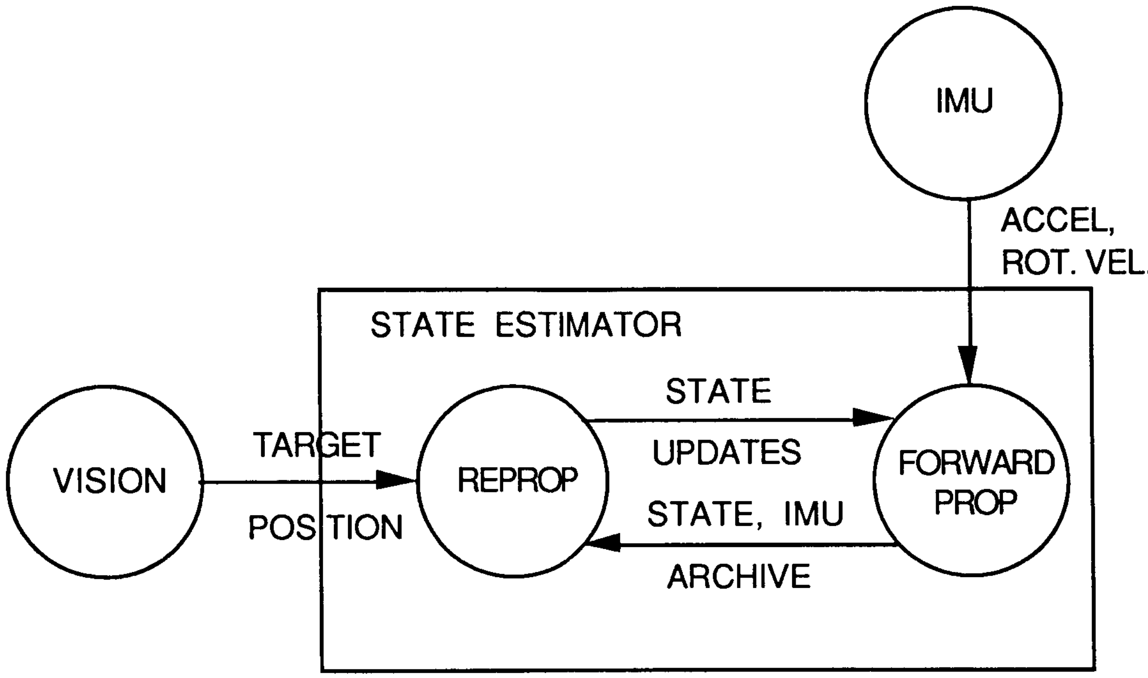


Fig. 4 Architecture of KC135 Translational State Estimator