Feature-Based Navigation.

J. Hallam, P. Forster, J. Smart, J. Howe

Department of Artificial Intelligence
Edinburgh University
5 Forrest Road
Edinburgh EH1 2QL
SCOTLAND
031-667-1011

Abstract.

The ability to navigate is an essential prerequisite for the construction of fully autonomous robotic vehicles. For underwater (as opposed to land) vehicles, the navigation problem is compounded by the continual perturbation of the vehicle motion by the movements of the medium. Continuous perturbation necessitates a continuous solution to the navigation problem: it is not possible to hold the vehicle stationary while the navigator computes where the vehicle actually is.

In this paper I describe a feature-based navigation system designed for the continuous navigation task (though it is also applicable in the land-based environment). Various versions of the system have been implemented, and have been tested using realistic simulations of suitable sensors. Experimental results derived from simulated two and three dimensional motion trials are presented.

The system infers the motion of a sensor-carrying vehicle from the observed movements of recognisable feature points in the environment. Feature points may be stationary or moving and may be only intermittently visible. Feature point motions need not be fully observable, though the performance of the navigator is best when full relative motion information can be inferred for each feature from its sensor returns. What does sensor return : Direction / Distance?

The system does not require *a priori* knowledge of the location or motion of the features on which navigation is based. It can also incorporate knowledge of vehicle dynamics if available and can utilise efficiently any direct vehicle positional or velocity information (such as rate of turn, heading, or odometry). What is odometry?

Acknowledgements: Part of the work described in this document forms part of a collaborative research project being jointly pursued with the Marconi Maritime Applied Research Laboratory, Cambridge UK, and funded under SERC Marine Technology Directorate grant GR/E/00068. Part of the work described was funded by SERC grant GR/C/44370. Facilities were provided by the University of Edinburgh.

Introduction

The ability to navigate is an essential prerequisite for the construction of fully autonomous robotic vehicles and a great variety of navigator systems have been devised for such vehicles. By navigation I mean the ability to know the relationship between the current position of the robot vehicle and its previous positions or the positions of objects observed in its environment. Navigational competence in this context is the ability to answer the question "Where am I?", rather than "How do I get to... ?".

In view of its importance, it is not surprising that a great deal of effort has been applied to the design and construction of effective navigation systems, mostly for application in land-based autonomous or semi-autonomous (*e.g.* guided vehicles in factories or warehouses) vehicles. The majority of these systems are based on *beacons* – readily visible objects which are fixed in the environment and with respect to which the navigational fixes necessary for computing vehicle motion may be taken. Beacon systems in general require that the beacons be fixed and stationary (for obvious reasons) and frequently need *a priori* knowledge of the positions of the beacons. Need for reference point.

Perhaps the earliest well known example of a beacon-based navigation system used by a mobile robot can be found in the work of Moravec (1980); similar techniques are proposed by Hannah ("bootstrap stereo", 1980) and used in modern missile navigation systems (e.g. Hostetler and Andreas, 1983). A good review of navigation systems for factory automatic guided vehicles can be found in (Cheng *et al.*, 1986).

The feature-based navigation System described in this paper differs from state-of-the-art beacon systems in two important ways.

- It does not require any *a priori* knowledge of the locations of features (analogous to beacons). Require here in project?
- It does not require the features to be stationary, and will determine automatically whether a feature is moving significantly.

In other respects it is very similar to the beacon navigation systems: the navigator uses knowledge of the positions and movement of features to infer the movement of the vehicle. Any direct vehicle motion information is also used in the inference process, and estimates of the position and motion of features are also obtained.

Motion Resolution

The feature-based navigation system operates by inferring the motion of a sensor-carrying vehicle (the observer) from the apparent motion of the observed environment. In order to do this, the observer must have access to a collection of *recognisable* points in the environment, which I shall call *features*, whose motion with respect to the observer can be determined.

These features, for which sensor-relative motions are supplied to the navigation system, may be any entity in the external environment which is recognisable over time and is visible fairly frequently (though not necessarily continually). The type of sensing used may be anything appropriate to the class of features considered, provided it can provide the apparent motion estimates required by the navigation algorithm. Examples of features and corresponding sensors include bright easily identifiable patches on marine objects observed with a sonar ranger, optical marker tapes viewed using a rangefinder, or pieces of environmental objects observed using a multistatic television camera array. Note that the beacons used by beacon systems are covered by this definition of a feature.

The basic problem addressed by the system is this:

> Given a sequence of noisy estimates of the apparent motion of a number of features in the environment, determine corresponding estimates of the motion of the vehicle and the features with respect to an arbitrary but fixed stationary frame of reference.

Effectively, the system is given motions of features relative to the sensor, and it determines which if any of the features have a proper motion, what their proper motion is, and what the motion of the sensor is. The global stationary reference frame is generated and maintained by the system as the 'frame of absolute rest' with respect to which all proper positions and velocities are recorded. It is arbitrary, but could be registered with an *a priori* world map if desired.

A formal mathematical description of the process, called ***motion resolution***, by which the apparent motions presented to the observer are resolved into their components is outwith the scope of this paper, but may be found in (Hallam, 1985). What follows is an informal description of the algorithm for linear observer motion and of the rotation compensation mechanism for dealing with observer orientation changes. ← observer changes direction.
↑ leave out.

Linear Motion Resolution

The linear motion resolution system deals with the motions of the observed feature points and with translational motion of the observer. It is structured as a computational cycle, outlined below. The flow of information during the cycle is summarized in Figure 1.

Motion Prediction

Needs initialization? How many cycles = previous

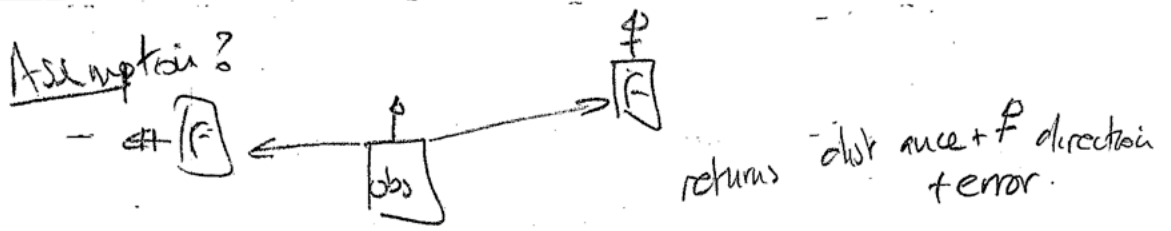
The current absolute position and velocity of each feature point and of the observer are predicted, based on estimates of their motions obtained from previous cycles. If a feature is believed to be stationary, its velocity for the purposes of prediction is set explicitly to zero. Modelled observer or feature dynamics may be included in this prediction process.

Feature Observation

New relative position and velocity estimates for each currently visible feature are obtained from the sensor(s). These estimates are either direct measurements or are the result

⇒ measurement + error

? angle, distance, orientation, ... etc. ?



of tracking filter computations on the direct sensor measurements. Each estimate is accompanied by an indication of its expected error covariance.

Reflected Motion

The relative motion measurement for each feature is subtracted from its predicted absolute motion. This difference is an estimate of the *reflected motion* (that part of the observed apparent motion which is due only to the movement of the observer) and is used to estimate the absolute motion of the observer. Again, an error covariance is associated with this estimate.

Observer Estimation

Estimates of observer motion derived by the previous step from the feature observations are combined, using a Kalman filter, with the predicted motion of the observer. If the observer's absolute motion can be wholly or partly measured by sensors, information from them can also be included in the estimation process. The result is a composite observer motion estimate which incorporates all the available information in proportion to the confidence expressed in it by its error covariance.

Feature Estimation

The composite estimate of observer absolute motion is added to the measured relative motion of each feature to provide an updated *a posteriori* estimate of that feature's absolute motion. The composite observer estimate and updated feature motion estimates form the basis for the motion prediction step in the subsequent computational cycle.

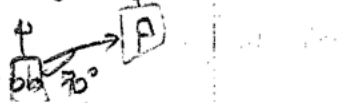
Motion Test

A motion hypothesis test is applied to the new estimates for each feature point to classify it as moving or stationary. If a feature is classified as stationary, its absolute velocity will be assumed to be zero during the prediction step – this is what enables the system to anchor its global reference frame to stationary features.

A feature point is included in the cycle whenever it is seen, and the prediction of its current absolute position and velocity is made with respect to the estimates computed at the last time it was seen. Feature points seen for the first time have no predicted absolute motion; the initialisation of the estimates for such features is discussed below.

Rotation Compensation

The main weakness of the linear system described above is its assumption that observer orientation is known. This is true either if the observer cannot alter its orientation during motion or if there is direct sensory information providing an estimate of the observer orientation at each point. If the latter obtains, then measurements made by the sensors can be corrected for the known varying observer orientation so that linear motion resolution takes place in a reference frame whose orientation is fixed. This correction process is rotation compensation.



Of course, the orientation of the observer will not in general be known perfectly. Further, the observer may not possess sensors able to provide imperfect orientation estimates. However, it is still possible to infer observer rotary motion and compensate for it, because the apparent motion of features will contain components due to the observer rotation and these will appear in the reflected motion estimates in the linear algorithm. They can be recognised because they are normal to the relative position of the feature.

Feature-based rotation compensation is achieved using a similar cyclic computation to the linear motion resolution system. It's schematic diagram is shown in Figure 2. The compensation mechanism is based on the following extra steps.

- Each time features are observed, the current observer orientation is predicted. The sensor measurements obtained from the features are compensated using the observer orientation prediction so that linear motion resolution may be carried out in a rotation compensated reference frame (*i.e.* one with nominally fixed orientation).

- The set of reflected motions computed are analysed for evidence of rotary observer motion and a composite estimate of the observer angular velocity is computed. Since the reflected motions have been calculated in a nominally fixed orientation frame, the rotation estimate is actually the error or *differential* observer angular velocity.

- The differential angular velocity is combined with any absolute orientation information available from direct sensing in order to provide the basis for subsequent observer orientation prediction.

- The feature velocities (relative and absolute) recorded by the linear motion resolution system are corrected for the differential angular motion. This correction is required because the angular velocity of the rotation compensated reference frame with respect to the world has decreased since the differential error has been corrected.

The observer differential angular motion actually induces non-linear feature motions in the rotation compensated frame of reference. Provided the differential angular velocity is small it is possible to obtain a good estimate of the angular motion. In this situation, it is possible to navigate entirely using feature-derived information.

Initialisation

As noted above, the motion resolving algorithm operates cyclically, combining predictions of feature point and observer motions based on previous cycles and current measurements obtained by sensors. In order for the cycle to begin, however, initial estimates of feature and observer motion are required. In this section I describe how those estimates are obtained.

The observer angular velocity is initialised to zero, since it can be shown that for moderate angular accelerations the compensation loop will converge geometrically to the correct estimate. This completes the initialisation of the rotation compensation mechanism.

$=\phi$

need feature & observer motion estimates.

The translational motion initialisation is accomplished using the first set of feature relative motion measurements obtained from sensors (note that this may imply several sightings of the features if their motion is not fully observable in a single sighting). The absolute positions of these feature points are set equal to their observed (relative) positions, they are assumed stationary, and the translational motion resolution algorithm is cycled once beginning with the Reflected Motion step. This initialisation is equivalent to assuming that the origin of the absolute reference frame is coincident with the current position of the observer. (If the absolute origin and orientation must register with an external map, for example, that registration may be achieved after the motion resolution initialisation has finished).

Since it is likely that some of the feature points seen initially will be moving, there will in general be classification errors in assuming that all are stationary. The misclassifcation of moving features causes the global reference frame to drift, and the drift continues until all moving features are correctly classified. Only misclassified moving features have this effect – if a stationary feature is misclassified the navigator estimates (typically) small proper motions for it.

All new feature points (apart from the initial set) seen during the operation of the algorithm are assumed to be moving, in order to avoid the potentially disastrous misclassifcation of moving features. Their estimated absolute motions are computed by applying only the Feature Estimation step at their first sighting.

Determining Feature Motions = Moving or Stationary.

The status (moving or stationary) of a feature point is determined each time it is seen using an appropriate hypothesis test. The hypothesis test may be either local, *i.e.* dependent only on the feature whose status is in question, or global, in which case the consequences of the local decision on other feature estimates is taken into account when the decision is being made.

In the two dimensional motion system a local hypothesis test was used. This was based on the sequence of estimated absolute velocities of the feature concerned, the classification method being a chi-square test operating on the estimated mean and error covariance of the velocity sequence.

In the three-dimensional system a more sophisticated global hypothesis test is employed. First described in (Hallam, 1987), this test is based on the notion of a weak assumption of a stationary environment. Features which are assigned moving status are in violation of this assumption and a fixed constant penalty is paid for each such feature. Features that have stationary status incur a cost computed from their estimated absolute velocity – clearly, a stationary feature should have a small estimated motion. The motion decision test is applied to all of the features observed during a cycle, and the status values assigned are those that minimise the global cost of the decision.

What is chi-square test?

Experimental Performance

Two experimental feature-based navigation systems have been implemented, one able to handle full two dimensional motion of the observer and one able (to date) to deal with translational and known rotational observer motion (*i.e.* it assumes perfect observer orientation sensing). In this section I present performance data for these two systems.

The Two Dimensional System

The algorithm has been implemented and has been tested extensively[1] using simulated data. Full details of the results of the tests can be found in (Hallam, 1985); a summary of those results is presented here.

The motion of the observer and of feature points was simulated using linear state transition models. The models incorporated deterministic and random forcing inputs. Feature point observations were generated intermittently, the chance of a successful sighting of each point being a simulation parameter, and measurement noise (in polar coordinates) was added to the computed relative position measurement.

All the noise vectors in the simulation were Gaussian and their covariances were simulation parameters. The implementation used Kalman tracking filters to estimate the velocity of each feature point from the intermittent sequence of noisy relative position measurements provided by the simulation for that point.

In all the tests reported below the measurement noise parameters were set to correspond to the performance of a typical marine sonar sensor. The radial resolution assumed was $6cm$ ($1cm$ noise standard deviation) and the angular resolution was 3.4° (10 milliradian standard deviation) and the sensor repetition rate was $2.5Hz$.

Two series of Monte Carlo tests comprising twenty members each were used to investigate the performance of the algorithm for composite rotational and translational observer motion. In all the tests the observer motion parameters were chosen at random from Gaussian distributions with $0.1ms^{-1}$ standard deviation and the set of feature points was uniformly distributed at random in a $200m$ square.

In the first set of tests no feature points were moving. Thus there was no danger of moving feature misclassification when the first set of feature points was seen. The results for this set of tests are presented in Table 1.

In all these tests the algorithm was successful in establishing a stable absolute reference frame. The frame spatial stability measure represents the degree of consistency between the simulated disposition of the feature points and the disposition estimated by the algorithm. This was assessed by fitting a least square error affine transformation between the true and estimated absolute positions; the quantity displayed in the table is the fit error per feature point.

In the second set of tests there were four stationary feature points arranged in a $70m$ square

[1]This work was funded by SERC grant GR/C/44730.

	Observer Motion					
	$\omega/mrds^{-1}$	v_o/mms^{-1}		σ_P^2/cm^2		Fit Error/cm^2
		x	y	x	y	
True	47.13	51.76	71.25			
Estimation Error	0.02	0.61	0.38	2.87	0.65	44.29
Max. in any test	0.70	6.51	2.69	352.30	66.85	786.10

	Feature Motion Estimation Errors				
	$\delta v_o/mms^{-1}$	R/m	σ_m^2/m^2	σ_P^2/m^2	
				x	y
max δv_o	11.75	101	1.02	0.0065	0.1283
min δv_o	0.04	99	0.62	0.0516	0.0233

R	Approximate range to feature
ω	Observer angular velocity
v_o	Observer linear velocity estimate
δv_o	Observer velocity estimation error
σ_m^2	Measured feature position variance
σ_P^2	Variance of position estimation error

Table 1: Stationary Feature Series: 20 tests, 4-10 visible features. The results of a typical test are shown; all tests were successful.

	Observer Motion					
	$\omega/mrds^{-1}$	v_o/mms^{-1}		σ_P^2/cm^2		Fit Error/cm^2
		x	y	x	y	
True	99.04	37.06	22.04			
Estimation Error	0.19	1.75	0.26	146.4	86.3	235.9
Max. in any test	0.31	10.21	3.29	2275.0	213.6	681.9

Feature Motion Estimation Errors (3/7 moving)				
	$\delta v_o/mms^{-1}$	v_f	R/m	σ_P^2/m^2
max δv_f	14.77	120.4	101	0.202
min δv_f	5.91	297.6	44	0.020

R	Approximate range to feature
ω	Observer angular velocity
v_o	Observer linear velocity estimate
δv_o	Observer velocity estimation error
δv_f	Feature velocity estimation error
σ_P^2	Variance of position estimation error

Table 2: Moving Feature Series: 20 tests, 4 stationary features, 1-5 moving features. The results of a typical successful test are shown; 8 tests succeeded.

and between one and six moving ones. The velocity components of the moving points were Gaussian variates with $0.2ms^{-1}$ standard deviation. In these tests at least one feature point was misclassified as stationary at the start of a run. The algorithm succeeded in eight of the tests and performed comparably to the first set of tests.

In two of the successful tests the algorithm correctly classified only three of the stationary feature points and estimated small random velocities for the other one. This resulted in greater observer velocity errors (about five times the typical error) and position noise variances (about 10 to 12 times larger) than for the other tests. This degraded performance was nevertheless satisfactory.

Table 2 gives the results summary for this set of tests. Figure 3 shows a scatter plot of the true and estimated absolute positions of the feature points and observer for one successful test in this series.

All the failures in the series were caused by a failure to discriminate correctly between stationary and moving features just after initialisation. An example of a failed test in this series is shown in figure 4 where it can be seen that the misclassification of feature 1 has resulted in erroneous estimates for the motions of all the other features.

There were two reasons for these discrimination failures. First, the proportion of moving feature points in the failures was high (typically 3 to 5 moving points) and the number of points in total was small (at most 10). The misclassification of a feature point as stationary was therefore able to affect the observer motion estimates sufficiently to force all other points to be interpreted as moving. Second, the hypothesis testing mechanism used in this implementation was simple and tended to misclassify points occasionally because of the effect of velocity estimation noise. Any moving point so misclassified could potentially cause failure.

The global hypothesis test implemented in the three dimensional system (currently the subject of experimental investigation) was designed to avoid the propagation of error apparent in figure 4. It achieves this by trading off the number of moving features hypothesised against the consistency of the resulting estimates.

The Three Dimensional System

The three dimensional feature based navigator[2] deals with the situation in which the observer orientation is known (though work is in progress to extend it to full three dimensional motion resolution). It has been tested using computer simulations similar to those described above for the two dimensional system.

In addition, further testing has been carried out using data obtained after signal processing from a realistically simulated experimental sonar device. This sonar measures range and azimuth angle, has a 360° scan with a beam width of about 4°, an elevation beam width of about 30°, a range window of $20m$ to $100m$, and a repetition rate of $1Hz$. Since the sonar sensor is effectively a planar one, a direct vehicle depth sensor is also provided.

The vehicle follows a roughly elliptical three dimensional trajectory at a speed of between $2ms^{-1}$ and $5ms^{-1}$, taking some five minutes to complete the circuit. Plan and elevation views of the trajectory are shown in figures 5 and 6, which show both the true vehicle trajectory and the estimated vehicle trajectory. It is immediately apparent that the trajectory estimated by the navigator matches closely the true trajectory (so closely that in places it is hard to distinguish them).

The performance of the navigator in estimating the vehicle motion as it traverses the circuit is also shown in figure 7, which illustrates the actual estimation errors at each point along the trajectory. The navigator estimates the vehicle position to a worst case accuracy of about four metres, which is approximately 1% of the trajectory major axis and a considerably smaller percentage of the distance actually travelled by the vehicle. The net error over one circuit is about $0.5m$.

The test data shown in figures 5 to 7 were obtained in a test situation where the orientation of the vehicle was varied in a realistic way, depending on heading and speed, as it traversed the circuit. Perfect measurements of the orientation were, however, supplied to the navigator since it is not yet able to handle full three dimensional vehicle motion.

[2]The three dimensional system is being investigated in the course of a collaborative research project, the industrial collaborator being the Marconi Maritime Applied Research Laboratory at Cambridge, UK. The project is funded by SERC Grant GR/E/00068.

Conclusions

A feature-based navigation system, similar to state-of-the-art beacon systems, has been described. By comparison with beacon systems it can:

- infer the motion of a vehicle from the apparent motion of features in a noisy environment where passive motion is present;
- determine whether features under observation are moving or stationary and deal appropriately with each type;
- function without any *a priori* knowledge of feature position or motion;
- incorporate vehicle dynamics or absolute motion sensing uniformly;
- incorporate information from a wide variety of feature detection sensors using standard statistical estimation techniques.

The system has been implemented in both full two dimensional and partial three dimensional observer motion versions and has demonstrated good navigation performance in these situations. The two dimensional system is able to navigate accurately using only information obtained from a simulated sonar sensor, while the three dimensional version uses an additional direct depth sensor to compensate for the two dimensional nature of the available sonar information.

References

Cheng *et al.*, 1986 Good review for factory automatic guided vehicles.

Cheng R M H, Courbet Y, Surpanceau M, Favreau P, Fahim A 1986; "Investigation of an automatic guided vehicle (AGV) driven by camera vision", Proc. Intelligent AUtonomous Systems, ed. Hertzberger L O, published by Elseveir, Amsterdam, 162-167.

Moravec, 1980

Moravec H P 1980; "Obstacle avoidance and navigation in the real world by a seeing robot rover", Stanford Computer Science Report STAN-CS-80-813.

Hannah, 1980

Hannah M J 1980; "Bootstrap stereo", Proc. AAAI Stanford CA, 38-40.

Hostetler & Andreas, 1983

Hostetler L D & Andreas R D 1983; "Non-linear Kalman filter techniques for terrain-aided navigation", IEEE Trans Automatic Control **AC-28**, 315-323.

Hallam, 1985 Motion Resolution

Hallam J C T 1985; "Intelligent automatic interpretation of active marine sonar", PhD thesis, Edinburgh University.

Hallam, 1987

Hallam J C T 1987; "Computational descriptions for interdisciplinary research in vision and cognition" in "Real Brains, Artificial Minds" eds. Casti J & Karlqvist A, Elsevier, 1987.

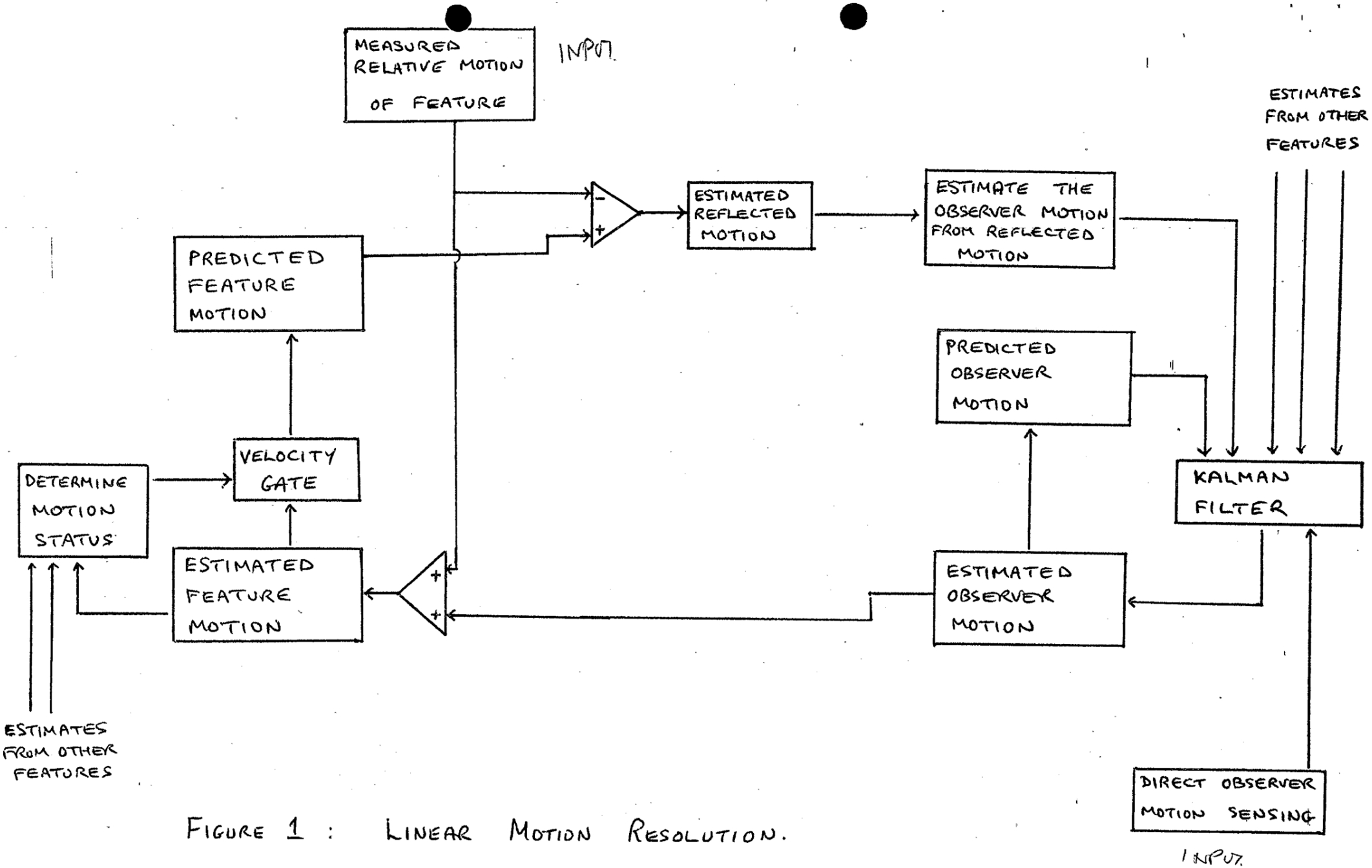


FIGURE 1 : LINEAR MOTION RESOLUTION.

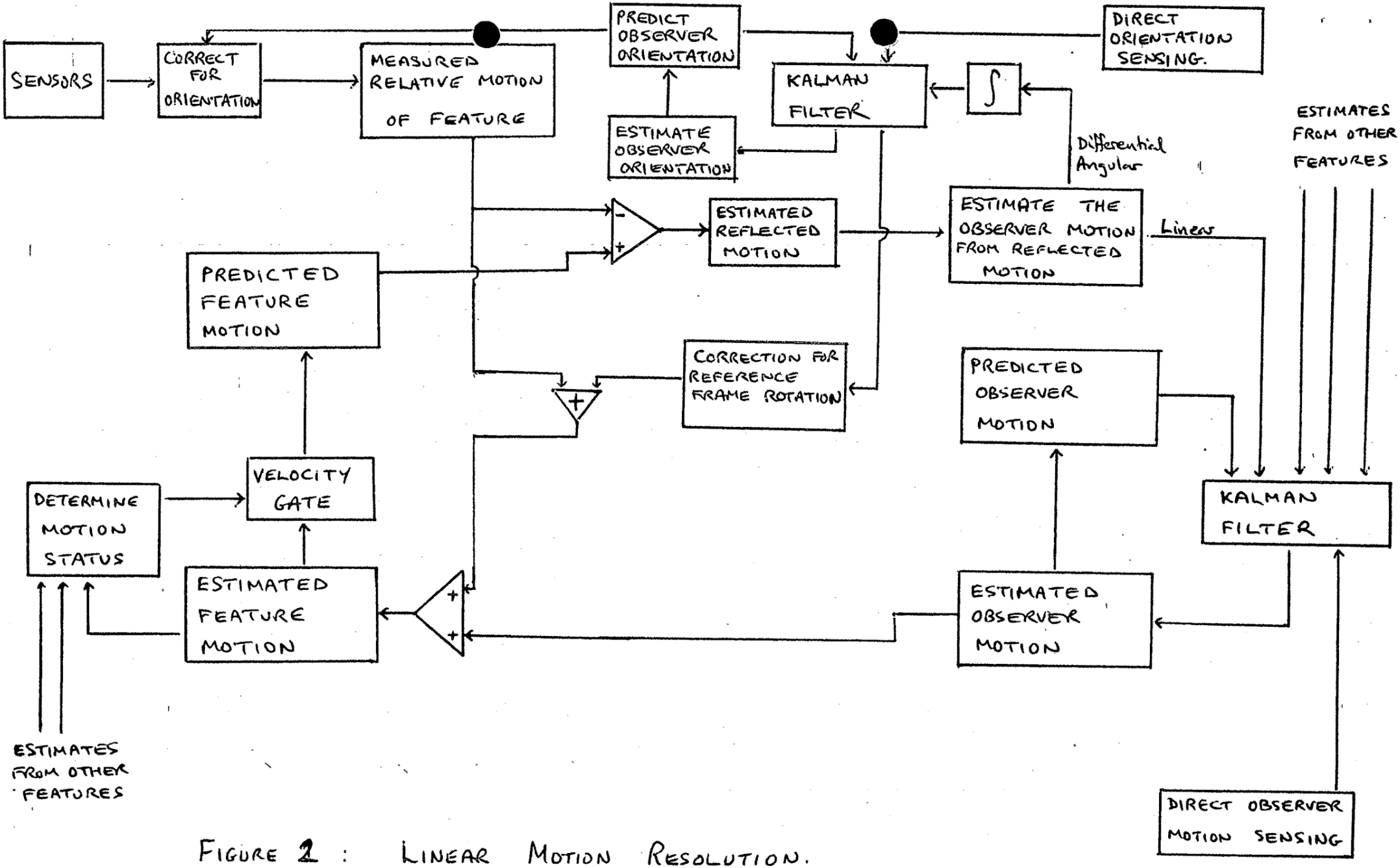


FIGURE 1 : LINEAR MOTION RESOLUTION.

WITH ROTATION COMPENSATION.

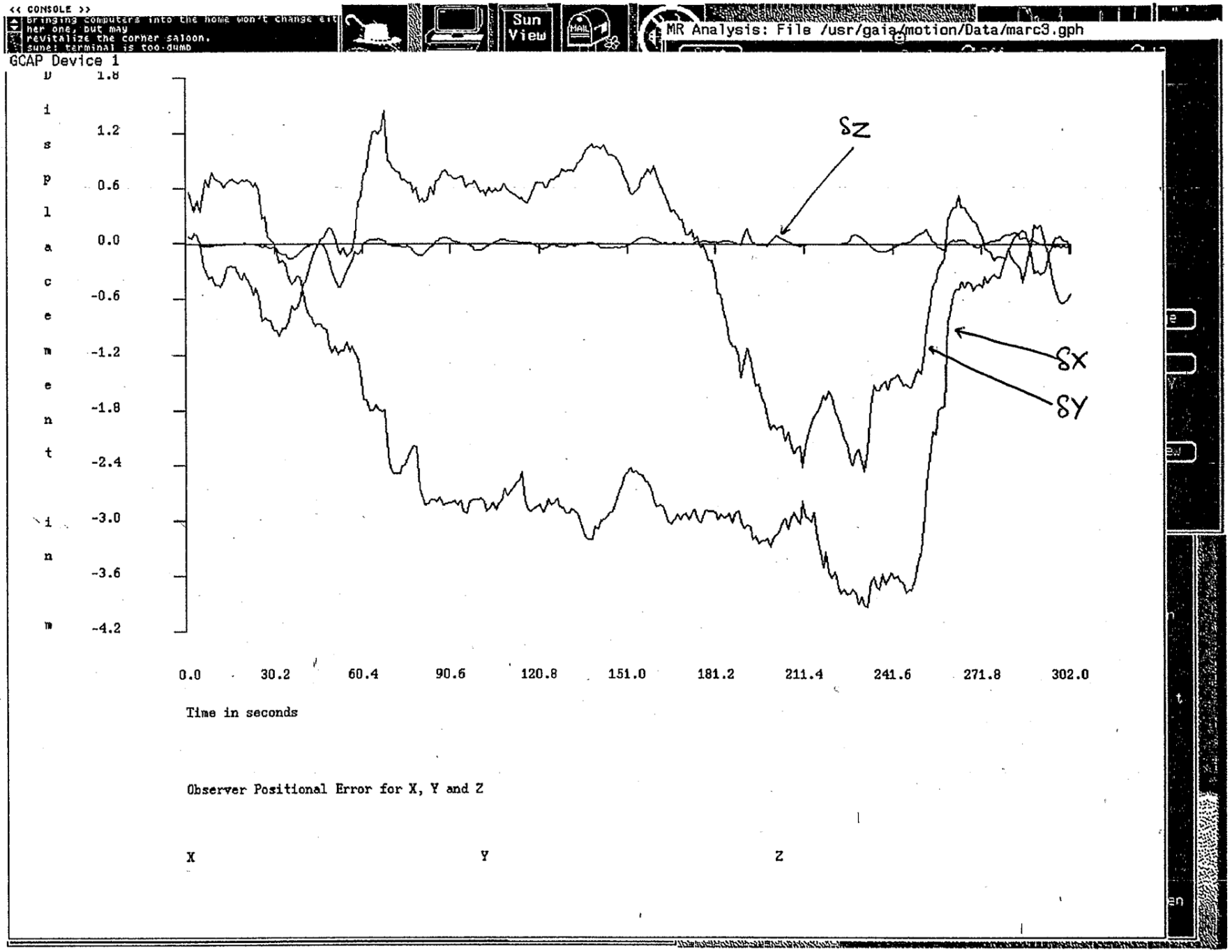

MR Analysis: File /usr/gaia/motion/Data/marc3.gph
GCAP Device 1
Displacement in mm
1.8
1.2
0.6
0.0
-0.6
-1.2
-1.8
-2.4
-3.0
-3.6
-4.2
Sz
Sx
Sy
0.0
30.2
60.4
90.6
120.8
151.0
181.2
211.4
241.6
271.8
302.0
Time in seconds
Observer Positional Error for X, Y and Z
X
Y
Z

Figure 7.

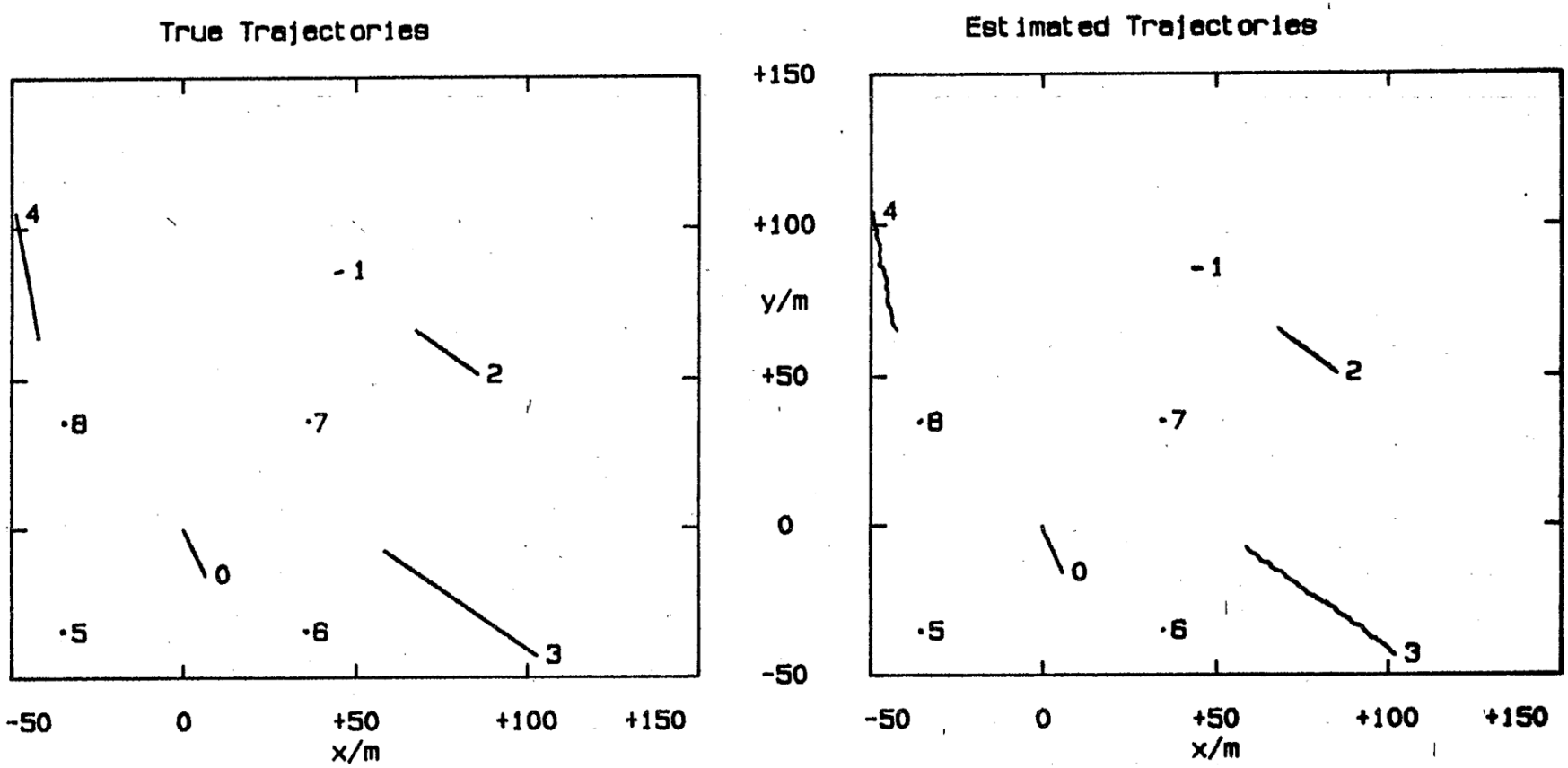
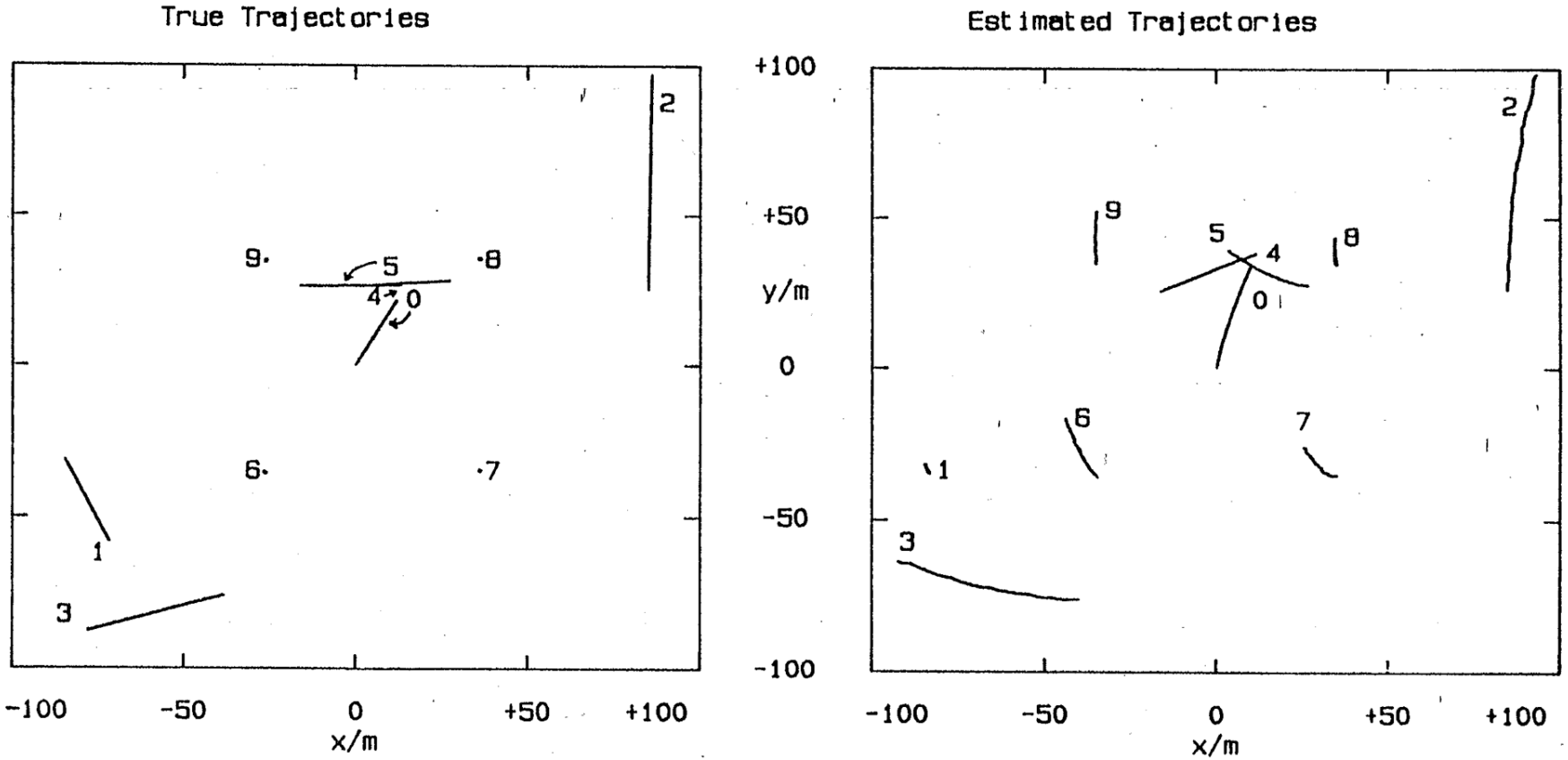


Figure 3. Scatter Plot of Absolute Target Positions for Test mn15.



Targets move towards their numbers: the observer is target 0.

Figure 4 Scatter Plot of Absolute Target Positions for Test mn5.

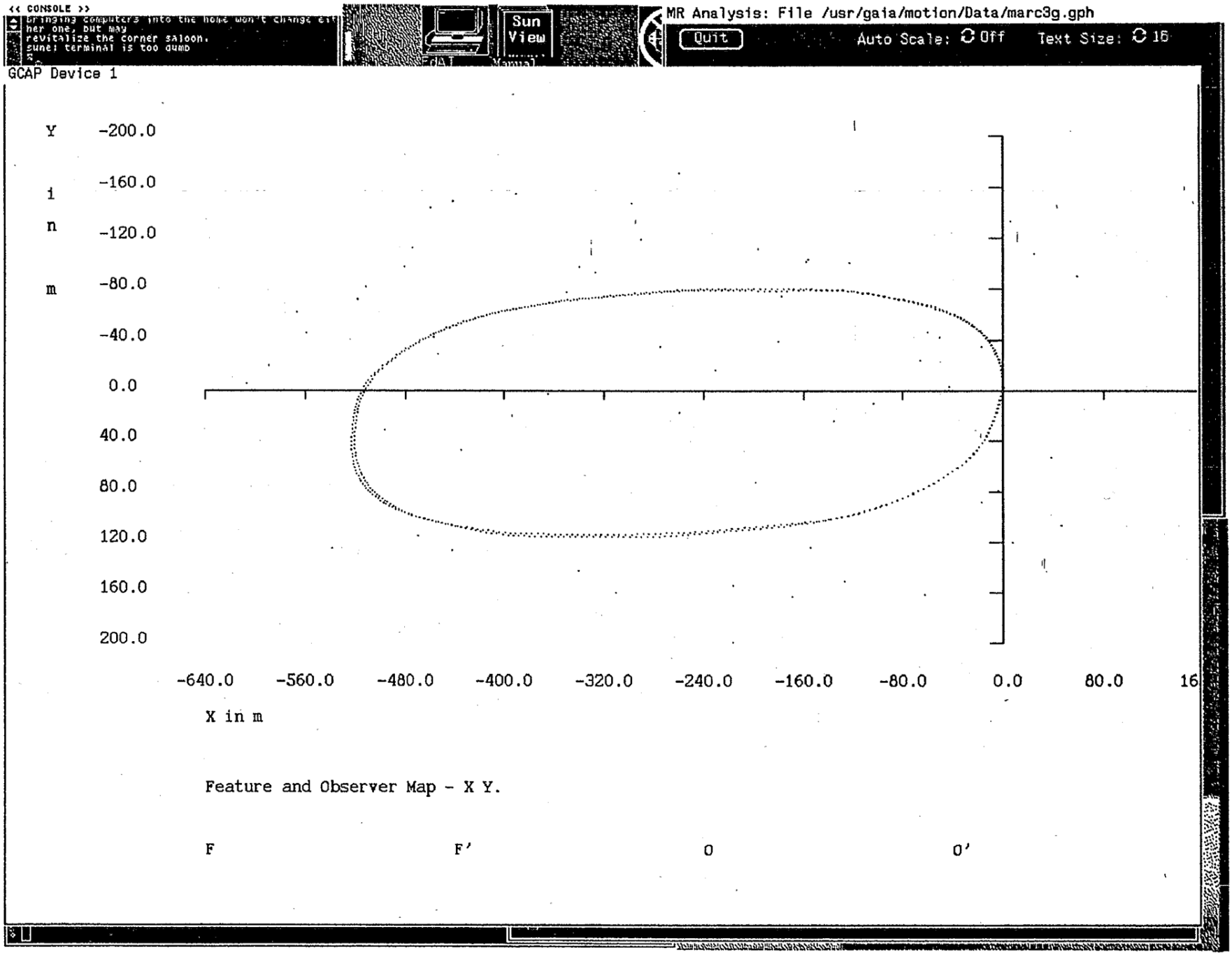


Figure 5. Test Trajectory plan

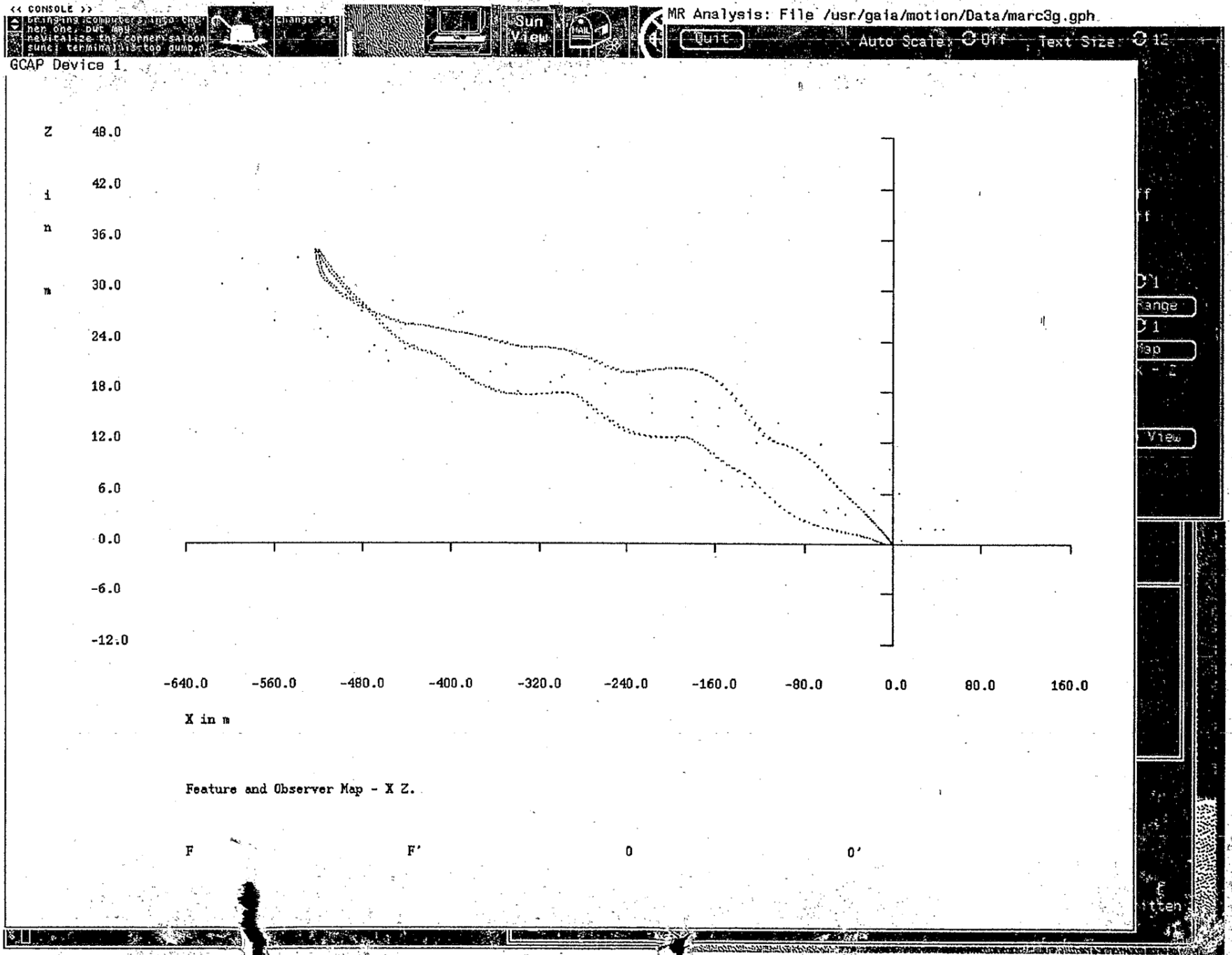


Figure 6 : Test Trajectory elevation.