Control and Estimation for Cooperative Manipulator Tasks

Lars Blackmore Steve Block

May 11, 2005

1 Motivation

The objective of this project is to achieve reliable transfer of an object from one robotic manipulator to another. This capability is useful for a number of applications, for instance robotic assembly, or robots with multiple manipulators, such as humanoid robots.

Achieving reliable object transfer poses a number of challenges for both control and estimation. As with most manipulation problems, the inverse kinematics problem must be solved so that the desired endpoint location can be specified in Cartesian coordinates, rather than in the joint space of the manipulator. An additional challenge particular to the cooperative robotics problem is that more than one manipulator may have a grasp on the same object. Manipulators that are carrying out simple position control may encounter problems when grasping the same object. Minor errors in forward kinematics can lead to large controller forces, or even unstable dynamics, as each controller tries to counteract the other to drive the perceived error to zero.

On the estimation side, carrying out reliable transfer depends critically on determining the *grasp state*; in other words, does a particular robot have a grasp on the object, or do both have the object? The grasp state must be determined before the sequence of events in a transfer task can proceed. For example, the manipulator receiving the object cannot move away until it is certain that the manipulator passing the object has released. In many instances, having pressure sensors mounted in the hand is infeasible. For example, packaging reasons can mean that the necessary space is not available, as is the case with the JPL LEMUR hexapod. We therefore need to *infer* the grasp state from the available observations, which are usually supplied by position encoders at the joints.

For this project we assume that each manipulator carries out estimation independently, without joint angle observations from the other robot, but with knowledge of its own joint angles and of the commands to be issued to both robots. This is typical of a multi-agent cooperative task, and the lack of observations makes the estimation task even more challenging.

This report describes the approach we use to solve this problem, which is comprised of the following two components.

1.1 Manipulator Control

We use *impedance control* to solve the inverse kinematics problem and to ensure that the manipulator is compliant to external forces. Layered above the impedance controller, we use a trajectory planner to continually update the desired endpoint position to produce smooth trajectories for movement across large distances and around obstacles.

1.2 Estimation

In order to solve the estimation problem we use a *Hybrid Estimation* approach, where we model the system as a Probabilistic Hybrid Automaton. Hybrid Estimation is able to infer the state of the manipulator, including the grasp state, using a model of the system dynamics, the observations, and the control inputs issued to the system.

Using two four degree of freedom manipulators, we implement our estimation and control approach and demonstrate that we can achieve reliable object transfer. We also show that hybrid estimation can be used to track the state of the system reliably. Finally, we demonstrate the ability of hybrid estimation to to detect faults, such as incorrect grasps.

2 Robotic Manipulator

The Model Based Embedded and Robotic Systems group in the Computer Science and Artificial Intelligence Laboratory has two Whole Arm Manipulators (WAMs), manufactured by Barrett Technology. A photograph of a WAM is shown in Fig. 1.

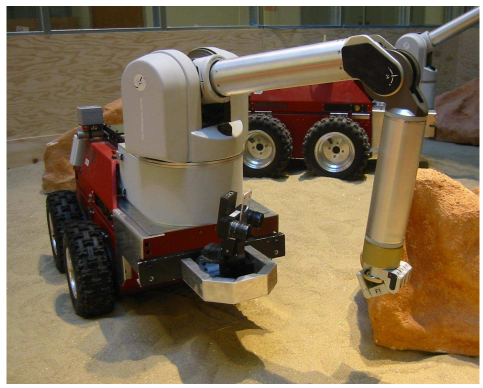

Figure 1: A Barrett Technology Whole Arm Manipulator (WAM).

The WAM consists of a two-link arm and a three-fingered hand. The arm has four degrees of freedom: two rotational joints at the base and another two rotational joints at the elbow. The hand also has four degrees of freedom: one provided by the curl of each of the three fingers and an other provided by the ability to rotate or 'spread' the fingers about the palm of the hand.

The WAMs are supplied with a software library that provides an interface to the manipulator. The library provides the user with the current joint angles and the WAM is controlled by setting commanded joint torques. These commanded torques are passed to an inner feedback loop implemented in hardware on the WAM which sets the motor torques. The algorithms described in this report were implemented in C++ and interfaced to the WAM through this library.

3 Manipulator Control

In cooperative activities such as the one considered in this project, where multiple manipulators must interact with each other, it is important that the manipulator offers a degree of compliance. This compliance means that the manipulator is able to tolerate unexpected external forces, which displace the arm from its desired configuration, without generating excessively large joint torques or causing failure of the control algorithm. In addition, for the manipulator to be of practical use, we must be able to specify the desired end effector location in absolute space. This requires that the control algorithm solve the inverse kinematics problem and calculate the corresponding joint angles.

3.1 Impedance Control

Our chosen control strategy is *impedance control*, which meets both of these requirements. Impedance control implements a virtual spring and damper between the end effector position $\mathbf{x}$ and the desired end effector position $\mathbf{x}_d$. Joint torques are calculated using the end effector position error $\mathbf{x} - \mathbf{x}_d$. The end effector position $\mathbf{x}$ can be calculated using forward kinematics and explicit calculation of the inverse kinematics is avoided because the transformation from end effector force to joint torques requires only the transpose of the manipulator Jacobian J.

In the case of a four degree of freedom manipulator such as the WAM, the manipulator has an extra degree of freedom with respect to setting the a three-dimensional desired end-point location. Therefore, specification of a desired endpoint location is insufficient to determine a unique set of joint angles $\mathbf{q}$. With impedance control engaged, an external force applied to the WAM at any point other than the endpoint will cause the links to move in such a way that $\mathbf{x}$ remains constant while $\mathbf{q}$ varies. Conversely, the set of joint angles $\mathbf{q}$ when the endpoint arrives at the desired location is a function of the initial value of $\mathbf{q}$ and hence the initial $\mathbf{x}_d$.

For the object transfer task, we wish to position the manipulator's hand in a particular orientation when picking up or transferring the object. This means that we wish to place a constraint on the joint angles $\mathbf{q}$ as well as on the end effector position $\mathbf{x}$. We apply this extra constraint by implementing PD control to drive joint angle q_3 towards a desired value.[1]

This is equivalent to impedance control on a compound state $\mathbf{a} = [x, y, z, q_3]^T$, which gives the following control law for the joint torques τ, where $\mathbf{g}$ represents the joint torques required to counteract gravity and $\mathbf{a_d}$ is the desired compound state. The matrices K_p and K_d are the proportional and derivative gains respectively. They represent the stiffness of the virtual spring and the damping coefficient of the virtual damper and allow us to modify the transient response of the manipulator.

$$\tau = \mathbf{g} - J^T \left(K_p \left(\mathbf{a} - \mathbf{a_d} \right) + K_d \dot{\mathbf{a}} \right)$$

Note that the dynamics of the endpoint under impedance control are not simple. The motion required of the manipulators links to achieve a certain endpoint motion means that the effective mass in the virtual spring and damper system is a complex function of the joint angles. Therefore, the dynamics are highly non-linear. However, for reasonable arm configurations, K_p and K_d can be used to qualitatively control the manipulator's dynamic response in a way similar to that which we would expect for a linear system.

3.2 Trajectory Planner

The impedance controller described above allows us to specify a desired endpoint location and K_p and K_d can be used to tune the dynamic response. However, this level of control is insufficient to produce reasonable trajectories for large movements, particularly if obstacles, self-impingement and singularities are to be avoided. We therefore implemented a trajectory planner which operates at a level above the impedance controller. This planner allows the user to specify a trajectory as a function of time and simply updates the desired end effector position accordingly in real time. This allows smooth trajectories in both space and time.

4 Estimation for Cooperative Manipulation

Sec. 1 describes a manipulation task that involves passing an object from one manipulator to another. This task is described in more detail in Sec. 5. In order to achieve this, it is essential to be able to estimate the state of each manipulator and its grasp. The grasp state is particularly important, as different portions of the task, such as moving away from the hand-over point, should not be attempted until successful transfer has occurred. The success, or otherwise, of the transfer can only be ascertained by determining the grasp state of the manipulator.

[1] Note that ideally we would like to specify the hand orientation directly, such as the angle between the local 'up' direction for the hand and the upward vertical. However, the transformation from this angle to joint angle, and the differentiation thereof to obtain the Jacobian, is far from straightforward. Joint angle q_3 is the rotation at the elbow along the axis of the upper arm link and approximates this angle well for most arm configurations. Specifically, the two are very close at angles near $\frac{n\pi}{2}$ for integer n, which are common angles of interest for manipulation tasks.

The state of the system can be represented conveniently using both continuous and discrete variables. Continuous variables represent the position and velocity of the manipulator in Cartesian space, while discrete variables represent the finite number of different grasp states of the manipulator. The overall system state is therefore a hybrid of continuous state $\mathbf{X}$, and discrete state, or *mode*, $\mathbf{M}$. *Hybrid Estimation* is a technique that can efficiently estimate the *hybrid state* of a system given a system model and observations. An overview of the technique is given in the following subsection.

4.1 Hybrid Estimation Overview

Hybrid estimation aims to determine a distribution over the current hybrid state given the inputs to the system and the observations y so far i.e. $p(\mathbf{M}_t, \mathbf{X}_t|\mathbf{y}_{1:t})$.

Central to the hybrid estimation technique are *hybrid models*. There are many alternative modeling formalisms, however for this project we use Probabilistic Hybrid Automata (PHA). An example of a PHA is shown in Fig. 2. This is a simple model of an actuator component. It can be seen that the actuator has two discrete modes, `ok` and `failed`. In each of these modes the system has a different set of equations describing the continuous state evolution. Note that these equations are discrete-time, and involve random noise processes that are used to model process noise and observation noise.

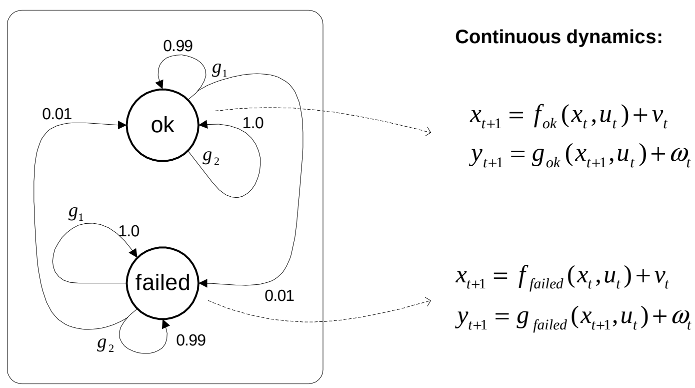

Figure 2: An example Probabilistic Hybrid Automaton (PHA).

The discrete state evolution is described by the transitions showed in Fig. 2. Depending on which of the guards g_1 or g_2 is satisfied, the probability of transitioning from one discrete mode to another at each time step is given by the probability shown in the model. The guards can be conditioned on inputs to the system or the state of the system. For example, it is possible to model systems where a command to the system will cause a particular transition to occur with high probability, but may alternatively cause an unintended transition with low probability. For the case of transition guards conditioned on the state, it is possible to model transitions that, for example, can only occur when the system's position is in a certain region.

The PHA modeling formalism therefore can be used to described many different systems, and is far more expressive than both purely continuous models, and purely discrete models. Hybrid estimation combines techniques from continuous estimation and discrete mode estimation to reason efficiently about these models. The approach is summarised here.

The hybrid state can be expressed as a sum of distributions over mode *trajectories*.

$$p(\mathbf{M}_t, \mathbf{X}_t|\mathbf{y}_{1:t}) = \sum_{\mathbf{M}_{1:t-1}} p(\mathbf{M}_{1:t}, \mathbf{X}_t|\mathbf{y}_{1:t}). \tag{1}$$

We can then expand each summand as a product of the probability of the mode trajectory $\mathbf{M}_{1:t}$ and a distribution over the continuous state, conditioned on this mode trajectory:

$$p(\mathbf{M}_{1:t}, \mathbf{X}_t|\mathbf{y}_{1:t}) = p(\mathbf{M}_{1:t}|\mathbf{y}_{1:t})p(\mathbf{X}_t|\mathbf{M}_{1:t}, \mathbf{y}_{1:t}) \tag{2}$$

A key observation is that, given a mode trajectory $\mathbf{M}_{1:t}$, the continuous system equations are fully specified. We can therefore use an existing continuous estimation technique, such as a Kalman Filter, to estimate the second term, $p(\mathbf{X}_t|\mathbf{M}_{1:t}, \mathbf{y}_{1:t})$. The first term can be calculated using the belief state update equation:

$$p(\mathbf{M}_{1:t}|\mathbf{y}_{1:t}) \propto P_O(\mathbf{y}_t) \cdot P_T(\mathbf{M}_t) \cdot p(\mathbf{M}_{1:t-1}|\mathbf{y}_{1:t}), \tag{3}$$

For details on the calculation of the observation function P_O and the transition function P_T please refer to [1] and [2].

Exact hybrid estimation would require us to track a Kalman Filter for every possible mode trajectory. However this is normally infeasible, since the number of mode trajectories increases exponentially with time. All practical hybrid estimation algorithms therefore carry out *approximate* hybrid estimation. This involves retaining only some of the possible mode trajectories; for example, in k-best enumeration, only the k trajectories with the highest posterior probability are retained. While this method approximates the true hybrid belief state, it is often possible to capture most of the belief state in a relatively small number of mode trajectories, leading to small approximation error.

In the following section we describe how the cooperative manipulation scenario can be modelled using the PHA formalism.

4.2 Modelling Cooperative Manipulation

In the cooperative manipulation scenario, we consider the case where each manipulator carries out estimation of its state using the observations from its encoders and full knowledge of the commands issued to both manipulators. In this case, the encoder readings, and therefore the state, of the other manipulator is *not* available. This is typical of many cooperative robotic scenarios where there is full knowledge of the planned task but sensor readings are not shared across agents. We consider hybrid estimation for WAM2, the recipient of the object.

In this scenario, the state, observations and commands are defined as follows:

- Continuous state: This consists of the position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ of the endpoint
- Discrete mode: This consists of the grasp state of the object M. In the mode `empty`, WAM2 does not have the object. In the mode `WAM2`, WAM2 has the object, and WAM1 does not. In the mode `both`, both WAM1 and WAM2 have the object.
- Observations: The observations are denoted $\mathbf{y}$ and $\dot{\mathbf{y}}$ and correspond to observations of $\mathbf{x}$ and $\dot{\mathbf{x}}$ from the encoders.
- Continuous commands: These are denoted $\mathbf{x_{d1}}$ and $\mathbf{x_{d2}}$, corresponding to the desired position for WAM1 and WAM2 respectively
- Discrete commands: u denotes the commands issued to the manipulators' hands. The possible values are `WAM1-OPEN`, `WAM1-CLOSE`, `WAM2-OPEN` and `WAM2-CLOSE`.

The discrete mode defined above describes three possibilities for the grasp state. Note that the `empty` state does not make any assertions about whether or not WAM1 has the object; without a grasp on the object, WAM2 has no information about the grasp state for WAM1.

While the continuous state of the manipulator is observed directly, subject to observation noise, the grasp state is not, since there are no sensors in the hand. The main aim of hybrid estimation in this context, then, is to estimate the grasp state, represented by the discrete mode of the system.

The inner-loop impedance controller has been used here to raise the modelling and estimation problem to a higher level of abstraction. While it would certainly be possible to perform hybrid estimation on the system at the level of torque inputs and raw encoder data, the modelling effort would be considerably increased. With an impedance controller, the endpoint dynamics can be described simply using known three-dimensional damping and stiffness terms. For a known grasp state, we consider the input to the continuous dynamics to be the desired positions $\mathbf{x_{d1}}$ and $\mathbf{x_{d2}}$, and the outputs are the true position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$. This is shown in Fig. 3.

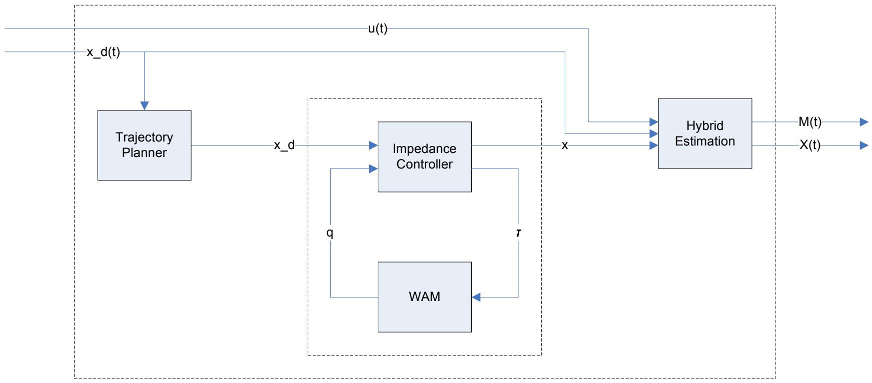

Figure 3: Schematic of the WAM control structure.

Furthermore, by using this inner-loop controller, the effect of modelling errors is reduced. For example, small errors in the gravity compensation term can lead to steady state controller errors that would not be anticipated by an estimation algorithm whose underlying model has gravity compensation and gravity effects perfectly balanced. By using relatively high proportional gains on the position error, these steady-state errors can be greatly reduced, reducing the effect of modelling errors.

4.3 Model Definition

In this section, the definition of the manipulator model is given by describing the continuous dynamics and outgoing transitions for each discrete mode.

4.3.1 Mode: `empty`

In this mode WAM2 moves freely without a grasp on the object. The input/output dynamics can be expressed in the form of a differential equation as:

$$\ddot{\mathbf{x}} = \frac{1}{m_2}\left[K_{P2}(\mathbf{x}_{d2} - \mathbf{x}) - K_{D2}(\dot{\mathbf{x}})\right] \tag{4}$$

This model assumes that gravity is compensated for perfectly, meaning that the only remaining terms are those due to the impedance controller. The use of the impedance controller allows us to model the endpoint without considering the configuration of the robot, as long as singularities are avoided. In the equation above, m_2 is the effective mass of the endpoint of WAM2. This mass is approximated as being constant, and was estimated by observing the dynamics of the system under impedance control in typical configurations. While this approximation is not insignificant in general, the value of the mass m_2 does not effect the steady-state behavior of the system. Hence during the low-speed tasks being performed in the scenario under consideration, this is a good approximation. A discrete-time approximation of this differential equation is used in the PHA model for the purposes of hybrid estimation.

The discrete mode transitions are as follows. The hand commands `WAM1-OPEN` and `WAM2-OPEN` clearly have no effect on the grasp state. In both of these cases, the discrete mode remains as `empty` with probability one. The command `WAM1-CLOSE` also has no effect; since WAM2 does not have a grasp on the object, it is unaffected by whether or not WAM1 has a grasp on the object or not.

If the command `WAM2-CLOSE` is issued, the hand on WAM2 closes. There are three possible outcomes of this action regarding the grasp state. Firstly, the hand can close but without a grasp on the object. In this case the grasp state remains as `empty`. Second, the hand can grasp the object, and the object is not being held by WAM1. In this case the grasp state transitions to `WAM2`, since now only WAM2 is grasping the object. In the third case, WAM2 grasps the object, which is currently held by WAM1. Now both manipulators have the object, and hence the grasp state transitions to `both`.

Table 1 shows the modeled transition probabilities for each discrete mode.

Command	`empty`			`WAM2`			`both`		
	`empty`	`WAM2`	`both`	`empty`	`WAM2`	`both`	`empty`	`WAM2`	`both`
WAM1-OPEN	1.0	0.0	0.0	0.0	1.0	0.0	0.1	0.1	0.8
WAM1-CLOSE	1.0	0.0	0.0	0.1	0.8	0.1	0.0	0.0	1.0
WAM2-OPEN	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
WAM2-CLOSE	0.8	0.1	0.1	0.0	1.0	0.0	0.0	0.0	1.0

Table 1: Transition probabilities

In the cases where the mode transitions are not certain, the exact probabilities were specified using modelling judgement.

Note that this transition model is consistent with the fact that WAM2 has no knowledge of WAM1 except the commands being applied to it. Without this knowledge, there is no way of knowing the result of the hand closing. Hence hybrid estimation must be used to infer the result from the observations.

4.3.2 Mode: `WAM2`

In this mode WAM2 moves freely, but has a grasp on the object. WAM1 does not have a grasp on the object. As a result the effective mass of the endpoint is increased, and in addition a force due to the weight of the object is exerted on the endpoint. The dynamics are now modelled as:

$$\ddot{\mathbf{x}} = \frac{1}{m_2 + m_{obj}} \left[-m_{obj} g \mathbf{e}_k + K_{P2}(\mathbf{x}_{d2} - \mathbf{x}) - K_{D2}(\dot{\mathbf{x}}) \right] \tag{5}$$

where $\mathbf{e}_k$ is the vertical unit vector.

In this mode the commands `WAM2-CLOSE` and `WAM1-OPEN` have no effect, and the discrete mode will stay as `WAM2` with probability one.

The command `WAM2-OPEN` will cause the hand to drop the object. This is modelled as having probability one.

The command `WAM1-CLOSE` will cause the hand of WAM1 to close. Again, since the location of WAM1 is unknown, the outcome of this action is unknown. The two possible results are, firstly, that WAM1 does not grasp the object, in which case the mode remains as `WAM1`, and second, that WAM1 does grasp the object. In the latter case the discrete mode transitions to `both`.

4.3.3 Mode: `both`

In this mode WAM1 and WAM2 both have a grasp on the object. The compliant nature of the impedance control means that this configuration is safe even if the desired positions of the two manipulators are not identical. The forces applied by the impedance controller on WAM1 are transmitted through the object to the endpoint of WAM2.

The resulting dynamics are:

$$\ddot{\mathbf{x}} = \frac{1}{m_1 + m_2 + m_{obj}} \left[(-m_{obj} g \mathbf{e}_k + K_{P2}(\mathbf{x}_{d2} - \mathbf{x}) - K_{D2}(\dot{\mathbf{x}}) + K_{P1}(\mathbf{x}_{d2} - \mathbf{x}) - K_{D1}(\dot{\mathbf{x}}) \right] \tag{6}$$

Here the effective mass has been increased by m_1, the effective mass of WAM1. The forcing terms are due to the weight of the object, the force applied by the impedance controller on WAM2, and the force applied by the impedance controller on WAM1, respectively. Note that we assume perfect gravity compensation for both WAM1 and WAM2, but the mass of the object is not compensated for.

In this mode, the commands `WAM1-CLOSE` and `WAM2-CLOSE` have not effect. In the case of `WAM2-OPEN`, the hand on WAM2 releases the object, and so the mode transitions to `empty`. This is modelled as occurring with probability one.

The most interesting case is that of `WAM1-OPEN`. In this case we assume that WAM1 releases its grasp with probability one. The expected outcome would be that WAM2 keeps hold of the object, and hence the mode transitions to `WAM2`. However experience showed that in a few cases, WAM2 did not have a firm enough grasp on the object, and that without the support of WAM1, the object fell through the hand of WAM2. Hence the model represents the fact that a `WAM1-OPEN` command can result in a transition to either `WAM2` or `empty`.

These definitions fully specify the model, given known values for m_1, m_2, m_{obj}, K_{P1}, K_{P2}, K_{D1} and K_{D2}.

5 Results

The tasks under consideration in this project is one where the first manipulator, WAM1, passes an object to the second manipulator, WAM2. Impedance control and trajectory planning as described in Sec. 3 were implemented for both manipulators. A trajectory was designed by specifying a number of way-points so as to enable transfer of the object. This trajectory can be summarized as follows:

1. WAM1 moves to pick-up point and grasps the object
2. WAM1 moves to the hand-over point with the object
3. WAM2 moves to the hand-over point
4. WAM2 grasps the object
5. WAM1 releases the object
6. WAM2 pulls away slightly to determine whether WAM1 has released successfully
7. If WAM1 released successfully, WAM2 moves to the the drop-off point
8. WAM2 releases the object

With this controller, successful object transfer was achieved in more than 95% of cases. Experiments demonstrated that compliance in the controllers was essential, as positioning errors of the order of a few centimeters were common. These steady-state errors came from a number of sources, including inaccurate encoder initialization, inaccuracies in gravity compensation, and friction. However the use of the impedance controller allowed the arm endpoints to comply when the grasp was made.

In the rare failure cases, excessive initialization error caused WAM2 to grasp the object poorly (e.g. at an angle), or to grasp WAM1's hand instead. Hence when WAM1 released, the object would fall. Typically however, these failures would be injected artificially as needed to demonstrate the hybrid estimation capabilities.

Hybrid Estimation was implemented for WAM2. Since the newly-acquired WAM hardware does not yet have the necessary architecture in place for the implementation of our hybrid estimation algorithms online, hybrid estimation was carried out off-line, using data acquired from the WAMs. In all of our experiments hybrid estimation tracked 20 mode trajectories and the hybrid update step took less than 0.1 seconds. This is the time step used in the dynamic model, so hybrid estimation could be used as a real-time estimator in this situation.

We present results for the following four scenarios.

- **transfer** The transfer is carried out as planned.
- **missed** WAM2 fails to grasp the object at the hand-over point, but carries on regardless.
- **stuck** WAM1 fails to release the object at the hand-over point, and the transfer is aborted.
- **slipped** Both manipulators grasp the object, but when WAM1 releases the object, it slips through WAM2's hand.

The results are shown in Figs. 4 through 7. In these figures, the top plot shows the continuous-valued information available to the hybrid estimator, while the middle plot shows the discrete-valued hand commands, also available to the estimator. The hand commands are issued over a period of approximately 1s, with a typical open or close operation lasting approximately 0.5s. The third plot shows the Most likely *A Posteriori* (MAP) discrete mode. The true discrete mode is also shown. While the estimation algorithm maintains a distribution over discrete modes at a given time step, the MAP estimate is the one often used in a control algorithm.

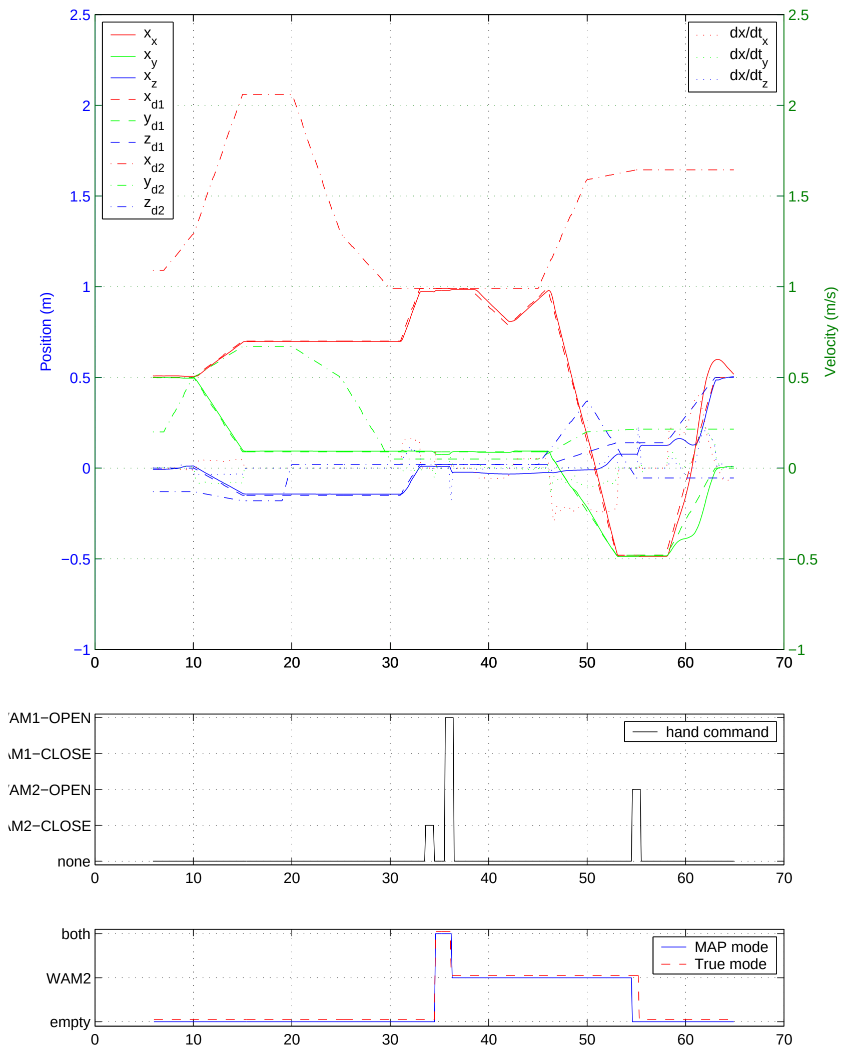

Figure 4: Hybrid estimation results for `transfer` scenario, with object mass of 2.050kg

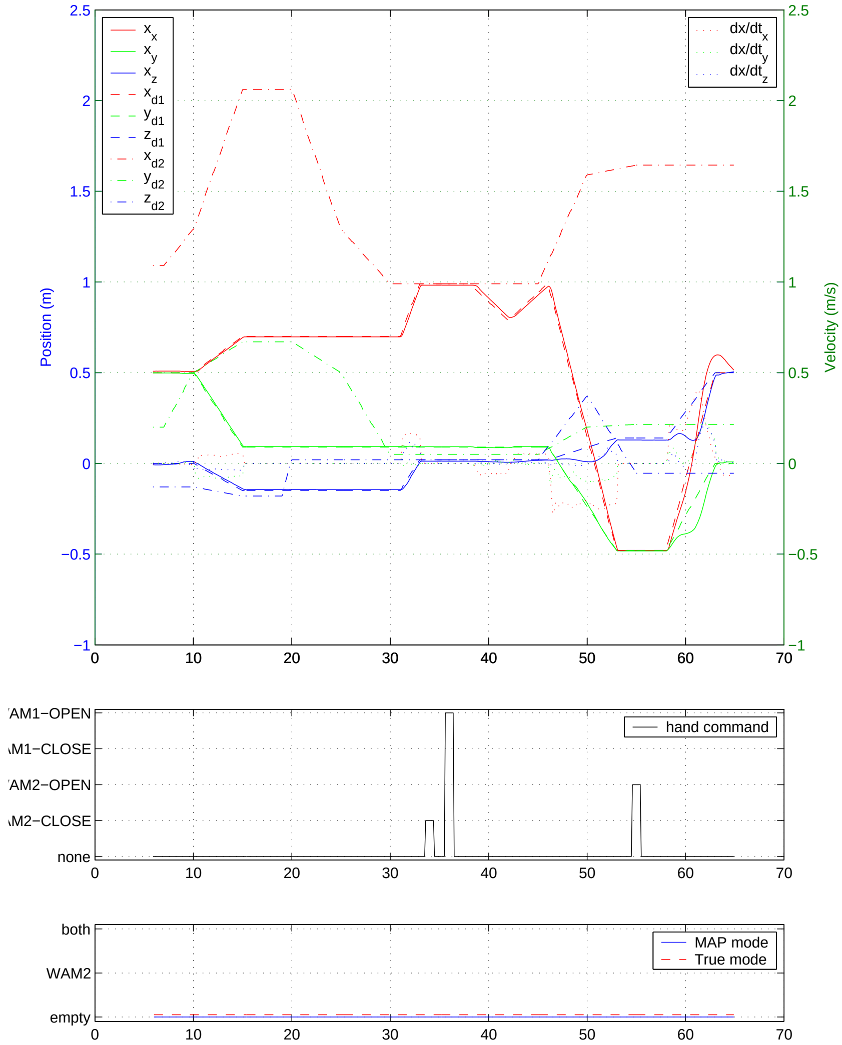

Figure 5: Hybrid estimation results for `missed` scenario, with object mass of 2.050kg

These results show that in all four cases, hybrid estimation tracks the true mode very accurately, with only minor differences in the timing of the mode transitions. Even in the `slipped` scenario, the most subtle of the four, the algorithm detects that WAM2 had a grasp on the object until WAM1 released, at which point the object slipped through the hands of both manipulators. The changes in discrete mode manifest themselves as relatively subtle changes in a set of complex and noisy observations, yet hybrid estimation is able to infer the discrete modes accurately by using the Probabilistic Hybrid Automaton model.

It would surely be possible to design an *ad hoc* estimator explicitly for detecting the presence, or otherwise, of an object in the manipulator's grasp. However the hybrid estimation approach is able to estimate the full state of the system in a probabilistically sound manner, by reasoning about the entire model. The only human design required was in the modelling phase, and creating accurate models is typically much more reliable than creating purpose-built *ad hoc* estimators. Furthermore, purpose-built estimators have been shown many times to lack robustness when used in situations that were not explicitly conceived of by the designer. We believe, therefore, that the hybrid estimation approach is a powerful tool for state estimation in this context.

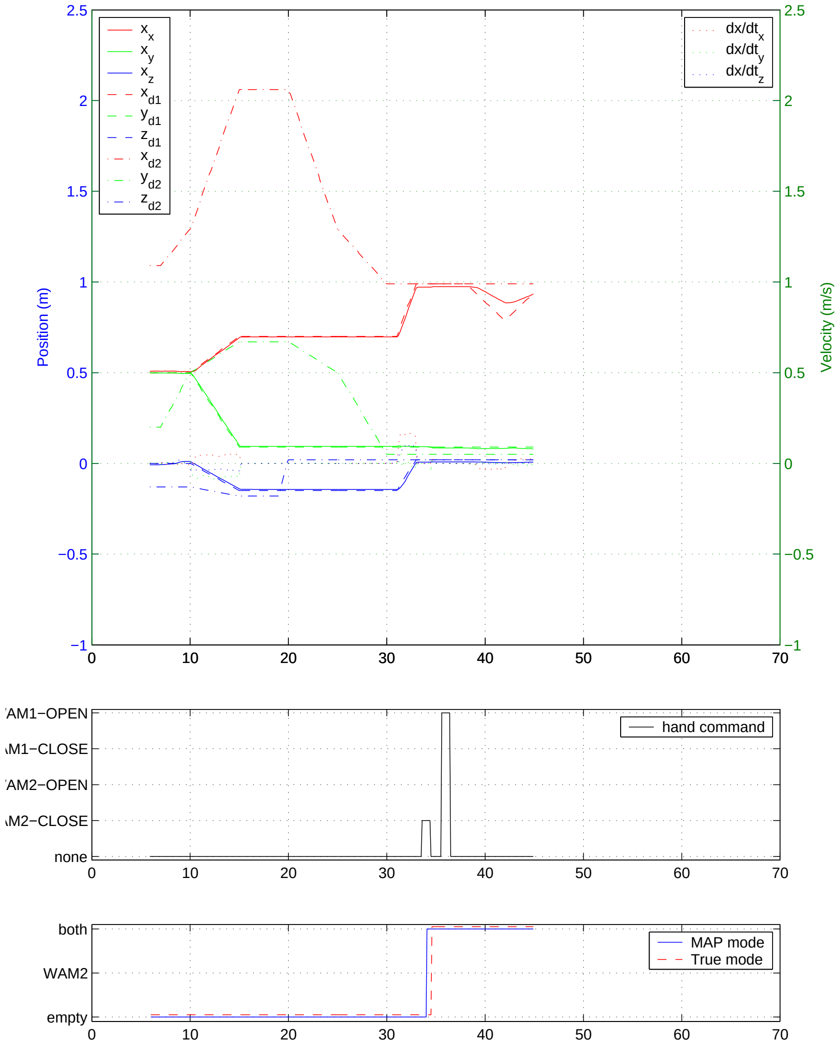

Figure 6: Hybrid estimation results for `stuck` scenario, with object mass of 2.050kg

5.1 Scenario Walk-through

To gain insight into the observation data available to hybrid estimation and the way in which it is used to track the state of the system, we present a qualitative walk-through of the data from the `transfer` scenario, shown in Fig. 4. Initially, both manipulators move very differently, so the desired position traces are very different, but the observed position of WAM2 follows its desired trajectory very closely. When stationary, the small error is due to inaccuracies in the gravity compensation parameters and to un-modelled joint friction. When the manipulator is moving, the system dynamics also produce a transient error. The system starts in the `none` mode and hybrid estimation can track this trivially, since no commands are issued.

At 33s, the trajectories of the two manipulators come very close together for the hand-over of the object. We see that WAM2 closes its hand at 34s and WAM1 then opens its hand at 36s. Between these two times the endpoint position of WAM2 is pulled between the two desired positions by the action of grasping the object and hybrid estimation uses this information to diagnose that the grasp mode is `both`.

Once WAM1 releases its grasp at 36s, the endpoint position of WAM2 moves back towards its desired position in the horizontal plane, but sags in the vertical direction due to the mas of the object, which is not

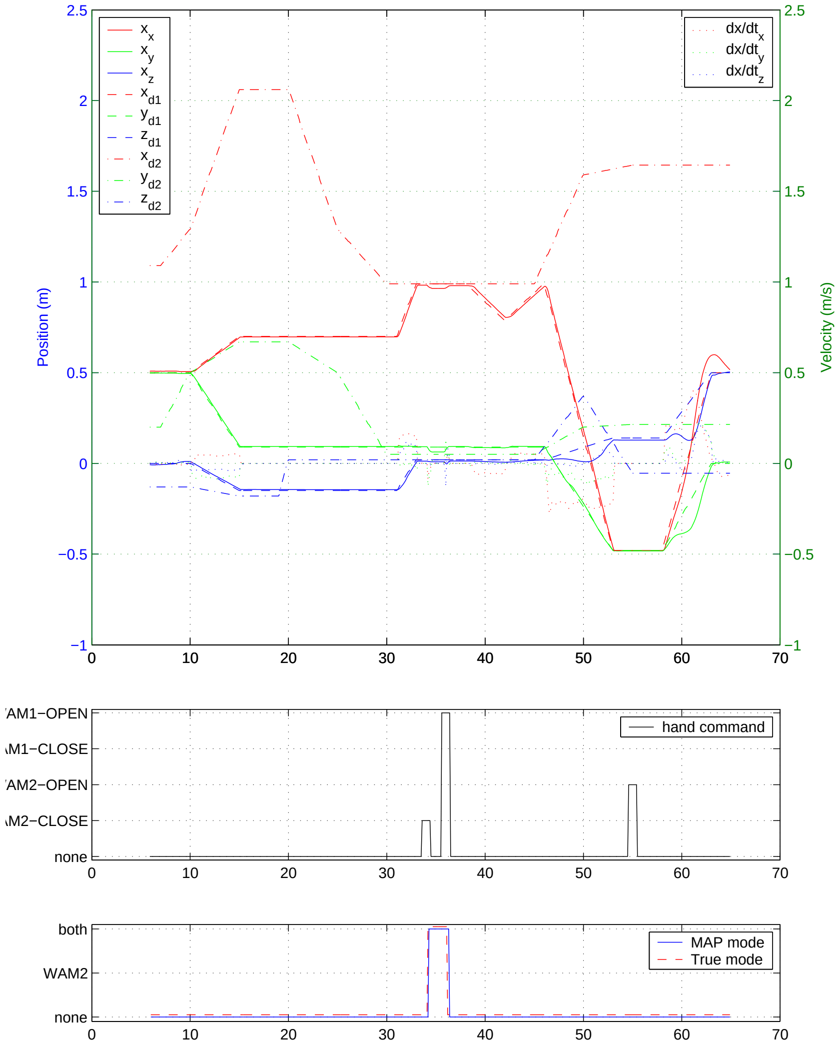

Figure 7: Hybrid estimation results for `slipped` scenario, with object mass of 2.050kg

accounted for by gravity compensation. This information allows hybrid estimation to determine that the grasp mode is now `WAM2`.

Between 38s and 46s, WAM2 performs an exploratory move to try to determine whether or not WAM1 has released its grasp on the object: we can see $\mathbf{x}_{d2}$ move away from $\mathbf{x}_{d1}$. In the case of the `stuck` scenario, where WAM1 does not release the object, the external force provided by WAM1 would prevent the endpoint of WAM2 from following its desired trajectory, as in Fig. 6. Here, however, this is not observed, so hybrid estimation can confirm that the grasp state is indeed `WAM2`.

At 55s, WAM2 opens its hand and releases the object. With this reduction in mass, the endpoint moves upwards towards its desired position and hybrid estimation is able to deduce that the grasp state has transitioned to `empty`.

6 Discussion and Analysis

In general, estimation algorithms are limited fundamentally by the amount of information available in the observations. For example, distinguishing the case where a very light object is grasped, from the case where the hand is empty will be extremely difficult, because the behavior of the system, and therefore the observations, will be almost unchanged by the presence of the object.

In particular, the signal to noise ratio of the observations sets the limit on the range of situations in which hybrid estimation can reliably track the correct mode. In the object transfer scenario, if the object mass is large compared to the virtual spring stiffness, the observation signal used by hybrid estimation to distinguish between the different discrete modes is large. In these situations hybrid estimation is very accurate.

The observation signal is noisy, however. If the magnitude of the noise is comparable to that of the observation signal (low signal to noise ratio), then hybrid estimation's task is very difficult. The noise in this signal originates from two sources: process noise and measurement noise. Hybrid estimation models both of these sources, but the model uses zero-mean Gaussian random variables. The manipulator has one significant source of process noise which is non-Gaussian. This is the steady-state error, which is due partially to inaccuracies in the parameter values used to calculate the gravity compensation torque, and partly due to joint friction, which is not accounted for in the control law. We can reduce the steady state error by increasing the stiffness of the virtual spring, which in turn reduces the un-modelled process noise. Therefore, to improve the chances of hybrid estimation producing a correct diagnosis, we should maximize the signal to noise ratio by maximizing the object mass, and maximizing the virtual spring stiffness.

The results in the previous section demonstrated that hybrid estimation can be used successfully with an object mass of 2.05kg. Based on the above analysis, we would expect the performance to degrade as the object mass is reduced and we would like to investigate this. Figs. 8 through 13 show the performance of the algorithm with objects of mass 1.175kg and 0.550kg.

When compared to the results with an object mass of 2.050kg, it is immediately clear that hybrid estimation performs better with large object masses. The trend is most obvious in Figs. 6, 10 and 13, which show the `stuck` scenario. Here the proportion of time for which hybrid estimation's MAP is incorrect increases as the object mass decreases. The true mode is initially `empty`, but transitions to `both` when WAM2 grasps the object at $t = 34s$. With the largest mass, hybrid estimation detected this transition almost immediately. As the mass is decreased, the observations are less informative, and hybrid estimation requires more evidence to detect the transition. This corresponds to an increase in the delay from the time of the true mode transition to when this is detected by hybrid estimation. Note that when WAM2 begins its exploratory motion at $t = 38s$, the evidence becomes more informative and hybrid estimation is soon able to detect the mode transition.

The results also confirm that the task of distinguishing the `WAM2` mode from the `empty` mode is significantly more difficult than that of distinguishing the `both` mode from the `empty` mode. Considering the `missed` and `stuck` scenarios, Figs. 5 and 6 show that with an object mass of 2.050kg, hybrid estimation performs perfectly. At 1.175kg and 0.550kg, the diagnosis is not perfect. At each of these two masses, hybrid estimation more accurately tracks the transitions between the `both` and `empty` modes in the `stuck` scenario (Figs. 10 and 13), than it does the transitions between the `WAM2` and `empty` modes in the `missed` scenario (Figs. 9 and 12).

Fig. 14 shows the variation in the percentage of diagnostic errors as a function of the mass of the object, for the `transfer`, `missed` and `stuck` scenarios. The percentage of diagnostic errors is determined by the number of time-steps during each trial that hybrid estimation's MAP mode did not match the true mode. The graph confirms that for each scenario, the performance of the hybrid estimator improves as the mass of the object is increased.

7 Conclusions

In this project we considered the problems of control and estimation for a multiple manipulator system. The scenario under investigation was one in which an object is to be passed from one manipulator to another.

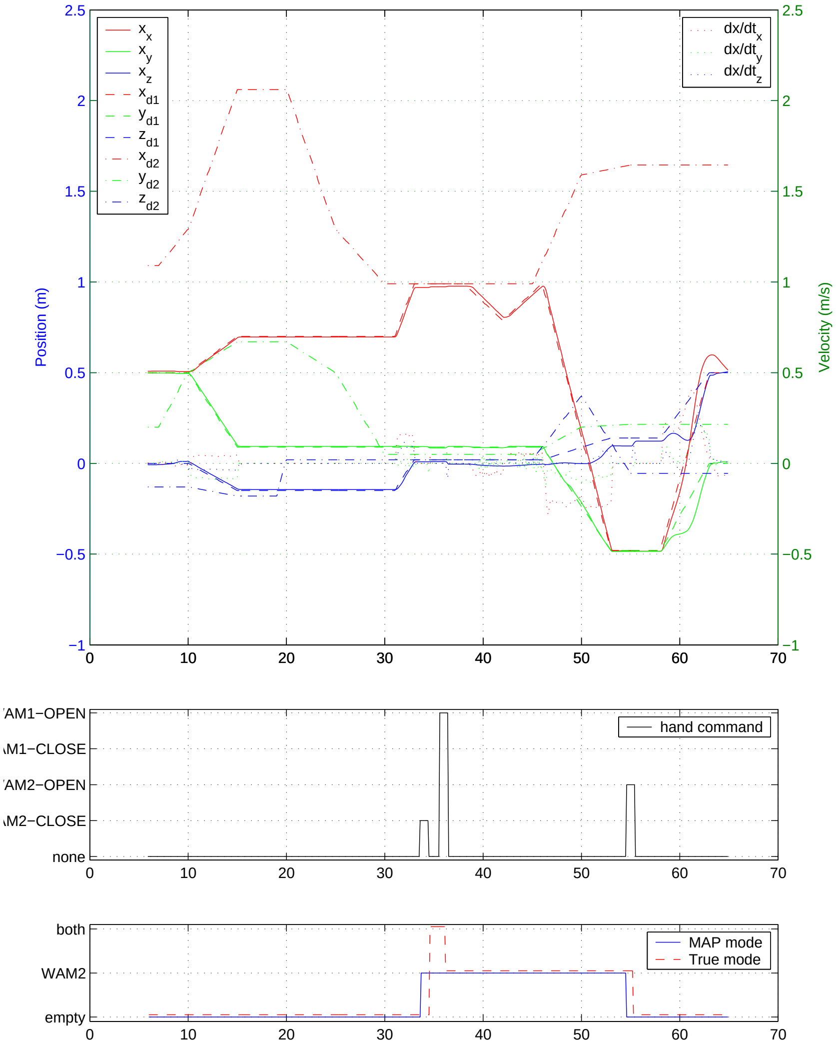

Figure 8: Hybrid estimation results for `transfer` scenario, with object mass of 1.175kg

An inner loop impedance controller provided compliance and inverse kinematics, while a trajectory planner was layered on top to provide smooth trajectory control. A hybrid estimation algorithm was used to track the hybrid state of the system and by reasoning on a physical model of the system was able to infer the grasp state, which was not directly observable. This architecture was implemented in hardware with a pair of 4-DOF manipulators.

Results showed that the system is capable of dependable transfer of an object from one manipulator to the other and that the hybrid estimator is able to reliably track the state of the system. In the case when failures occurred, the estimator correctly detected the failure mode and could be used to provide feedback to initiate recovery.

The effects of reducing the information available to the estimator, by lowering the mass of the object, were also investigated. Results showed that the percentage of diagnostic errors increased as predicted.

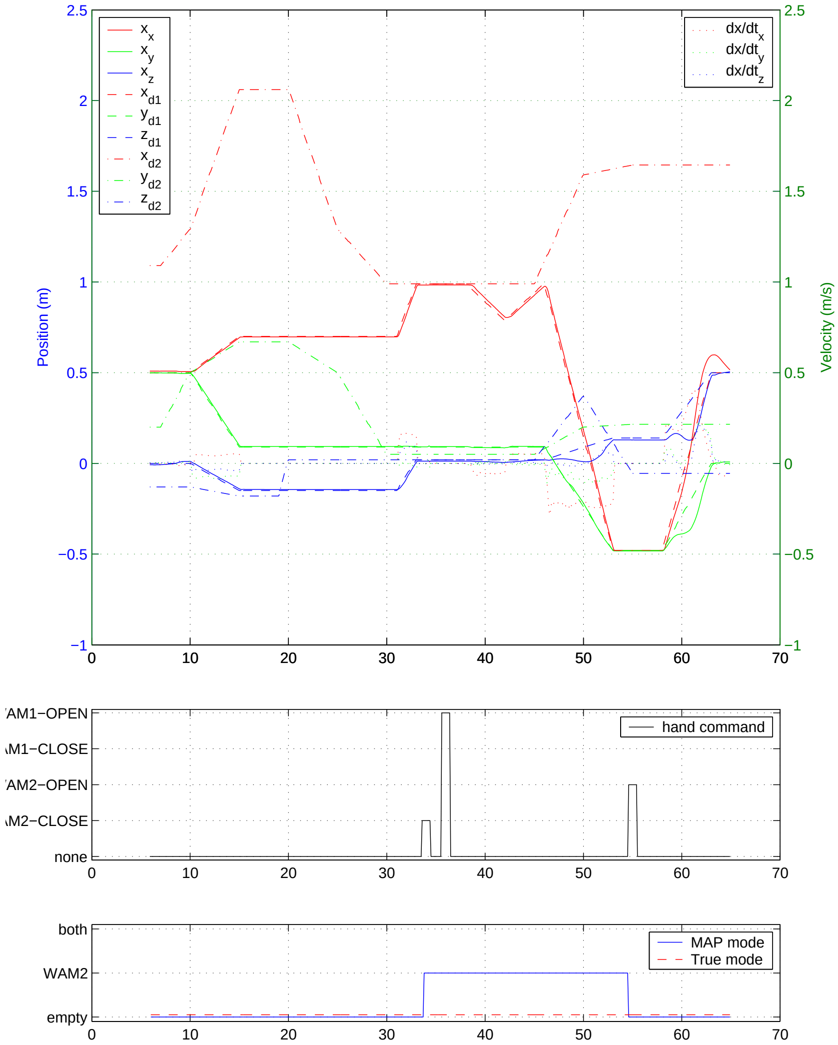

Figure 9: Hybrid estimation results for `missed` scenario, with object mass of 1.175kg

References

[1] M. W. Hofbaur and B. C. Williams. Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2004.

[2] M.W. Hofbaur. *Hybrid Estimation and its Role in Automation*. Habilitationsschrift, Faculty of Electrical Engineering, Graz University of Technology, Austria, September 2003.

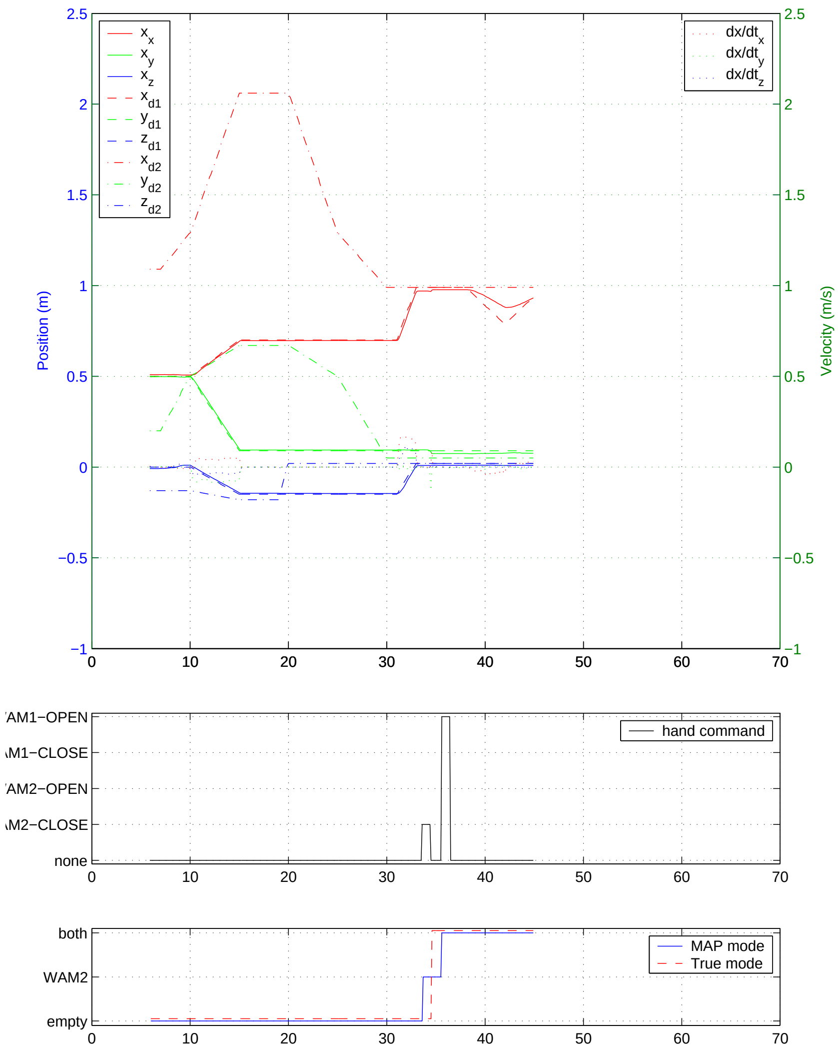

Figure 10: Hybrid estimation results for `stuck` scenario, with object mass of 1.175kg

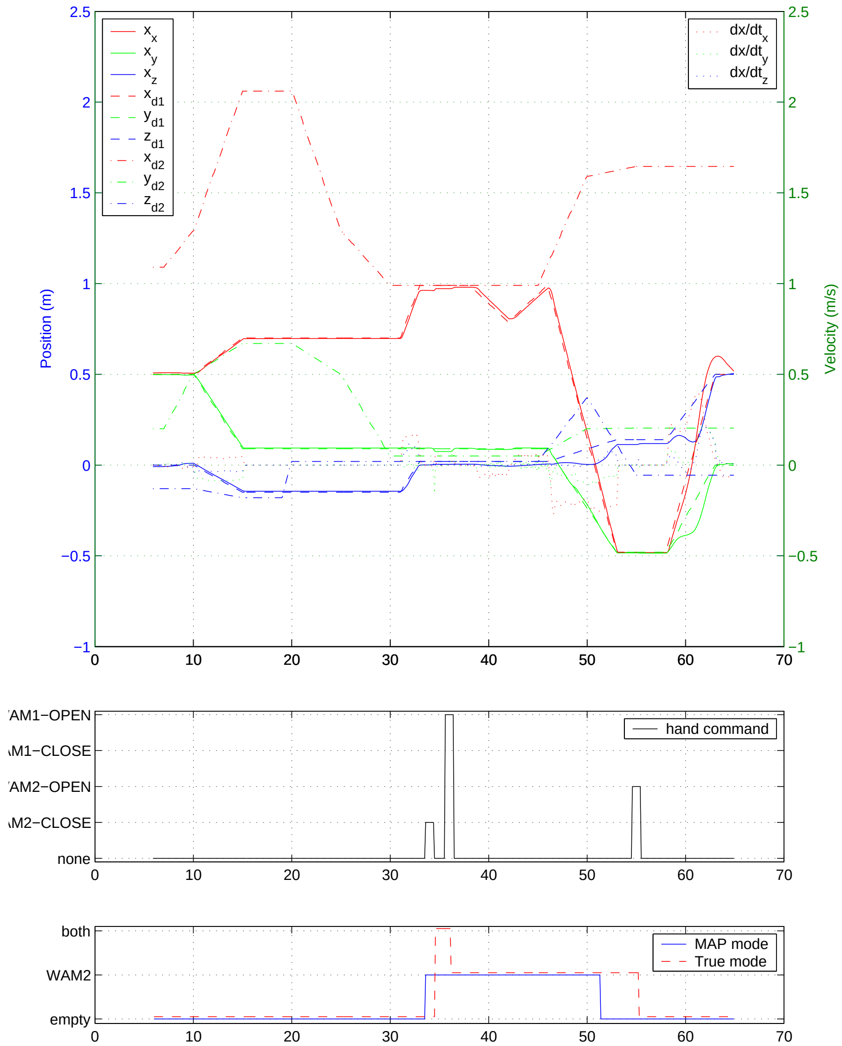

Figure 11: Hybrid estimation results for `transfer` scenario, with object mass of 0.550kg

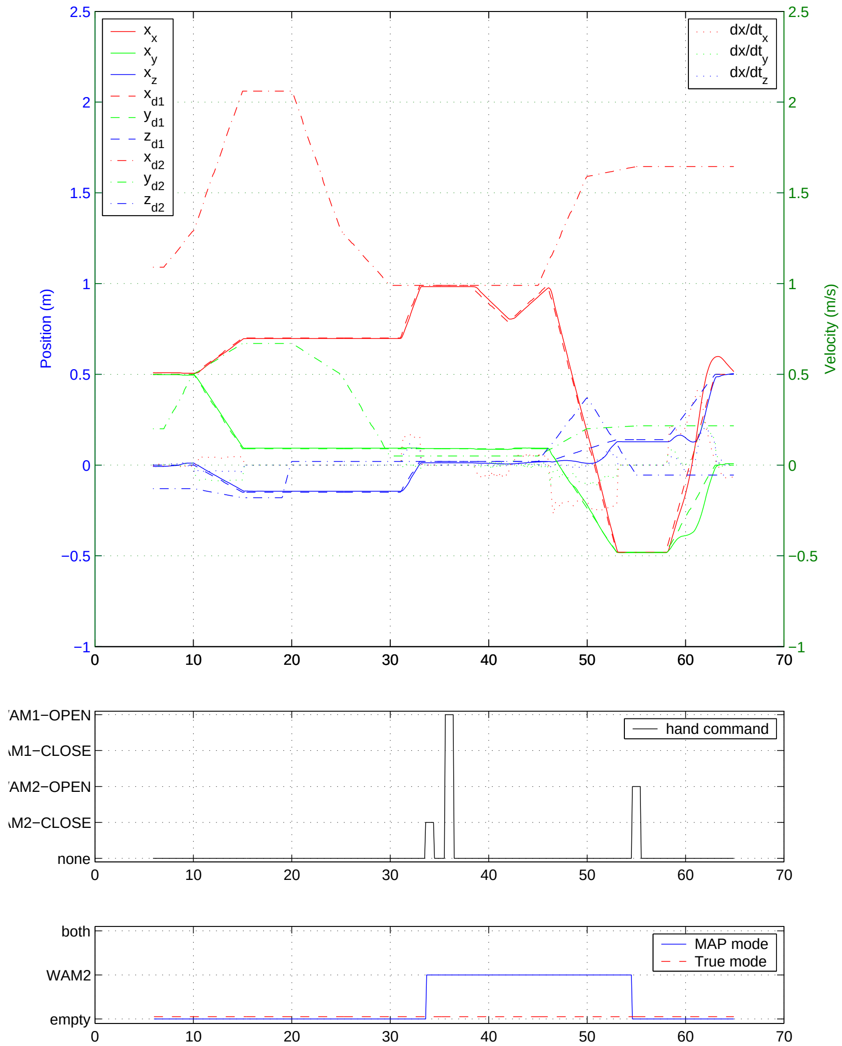

Figure 12: Hybrid estimation results for `missed` scenario, with object mass of 0.550kg

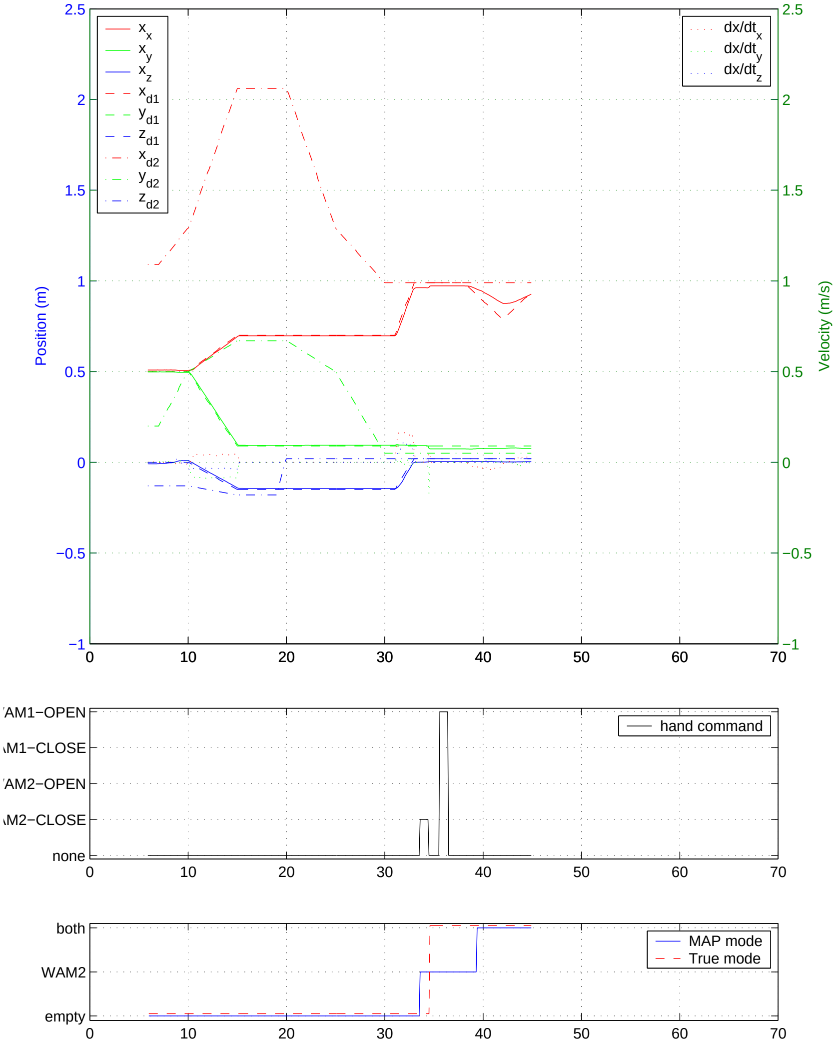

Figure 13: Hybrid estimation results for `stuck` scenario, with object mass of 0.550kg

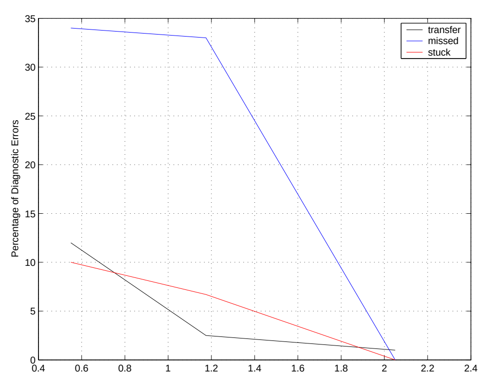

Figure 14: Performance of hybrid estimation as a function of object mass.