

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/50386942>

Robust autonomy for the youBot

Article

Source: OAI

CITATIONS

0

READS

115

1 author:

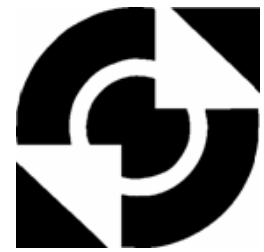


Douwe Dresscher

University of Twente

11 PUBLICATIONS 6 CITATIONS

SEE PROFILE



Robust autonomy for the youBot

Douwe Dresscher

MSc report

Supervisors:

prof.dr.ir. S. Stramigioli
dr.ir. P.C. Breedveld
MSc Y. Brodskiy

March 2010

Report nr. 003CE2011
Control Engineering
EE-Math-CS
University of Twente
P.O.Box 217
7500 AE Enschede
The Netherlands

Abstract

In line with developing the best practise in robotics(BRICS), this project implements one of the most advanced approaches to increase robust autonomy for mobile manipulators. Using the newly announced research platform youBot as an example, a robust controller with a protection layer against joint and wheel failures was developed. The protection layer consists of a set of algorithms implemented in software that is responsible for error detection and system recovery. The system includes a model-based fault-tolerant robust control algorithm to deal with transient faults and a reconfiguration algorithm for permanent failures. In addition to the development of the control algorithm, a walk-trough tutorial on the modeling of a complex robotic system in a reusable object-oriented style is presented. Simulation results show that the proposed system is successful at increasing the robust autonomy for sensing and actuation faults.

Contents

Abstract	iii
1 Introduction	1
1.1 Goal	1
1.2 Outline	1
2 Modeling of the youBot	3
3 Fault tolerant control	22

1 Introduction

Robust autonomy is increasingly important in modern autonomous robots. It is the ability of a system to react on both explicitly-specified adverse situations and unexpected adverse situations in both the environment and the underlying system (hardware and software) without or with a limited human guidance in order to complete its mission in the best possible (acceptable) way. It is the potential of an intelligent man-made mechanism for detection and reasoning about adverse situations without human intervention. The robustness of robot autonomy should be such that interactions with the environment based on reasoning remain sufficiently safe, accurate, etc.

Robust autonomy solutions could be viewed as features included in robot - resources that ensure "the happy flow" of its operation. Utilizing or introducing some type of redundancy into the system could achieve this. The only time when an autonomous robot can be provided with resources to achieve robust autonomy is at design time.

1.1 Goal

This project aims at increasing the robust autonomy of a new robot, youBot, by inserting a protection layer against hardware faults. The youBot is a mobile manipulator consisting of a robotic arm and a mobile base. The protection layer is focused at faults in the robot effectors viz. failures in the actuation of the arm or in the mobile base. The protection layer consists of a set of algorithms implemented in software that is responsible for error detection and system recovery. The system includes a model-based fault-tolerant robust control algorithm for transient faults which can be reconfigured in case of permanent failures.

1.2 Outline

This project is divided in two parts, which have been documented in two separate papers. The first presents a walk-trough tutorial on the modeling of a complex robotic system in a reusable object-oriented style. The second part covers the development of the model-based fault-tolerant control algorithm.

2 Modeling of the youBot

Modeling of the youBot in a serial link structure using twists and wrenches in a bond graph

Douwe Dresscher, Yury Brodskiy *, Peter Breedveld *, Jan Broenink *, and Stefano Stramigioli *

University of Twente.

Abstract. We present a walk-through tutorial on the modeling of a complex robotic system, like the newly developed desktop mobile manipulator youBot developed by KUKA[5, 4]. The tutorial shows the design of models for typical robotic elements, done in a reusable object-oriented style. We employ an energy-based approach for modeling and its bond-graph notation to ensure encapsulation of functionality, extendability and reusability of each element of the model. The kinematic representation of mechanical elements is captured using screw theory. The modeling process is explained in two steps: first submodels of separate components are elaborated and next the model is constructed from these components.

1 Introduction

The modeling of the dynamics of a complicated robotic system is a time consuming process. One of the ways to increase speed of development is reuse. This concept is rarely applied in modeling of robotic systems, because the mechanical parts of a robotic systems often require development of the dynamic models from scratch. Moreover, different applications require different types of modeling assumptions, even for the same robot. The concept of power-based modeling creates the possibility of representing a dynamic model in a simple way, while preserving the information about modeling decisions. This allows simple extension of the model. A bond graph is a graphical representation method of power-based modeling. Bond-graph models can be reused elegantly, because bond graphs are non-causal. The submodels can be seen as objects with defined power interfaces; bond-graph modeling is a form of object-oriented physical system modeling.

Bond-graph theory and notation are well developed and described in [3, 2]. In short, the representation is done by means of a directed graph: the vertices are submodels that describe interesting physical behavior and the edges represent energy relations. In addition to these edges, signals are used to pass information between vertices and do not represent energy exchange in the system. Bond graphs are a domain-independent representation of the model, therefore allowing easy capture the behavior of robotic systems, which often combines electrical,

* The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-231940-BRICS (Best Practice in Robotics).

mechanical and hydraulic domains. In this paper, we will only concentrate on the mechanical behavior and only indicate how the model can be extended to other domains.

In order to develop a well structured model, the system behavior should be decomposed into concepts, idealized descriptions of physical phenomena. The concepts are combined into recognizable dominant behaviors of the tangible system parts(components). The components are used to compose the model of the robotic system.

The dynamic behavior of the youBot (mobile manipulator robot) can be decomposed, without losing generalization, into three types of components: Mecanum wheels, joints of the manipulator and links of the manipulator.

Section 2 gives a description of the youBot. Sections 3.1 and 3.2 provide descriptions of the modeling assumptions and the modeling process for the links and joints of the manipulator using screw theory. Section 3.3 elaborates modeling of the Mecanum wheels. Section 3.4 presents the model composition process.

2 Description of the platform

For this paper, a robotic manipulator consisting of a robotic arm and a base driven by four Mecanum wheels is used as an example. The kinematic structure of the youBot robotic manipulator is shown in Fig. 1b.

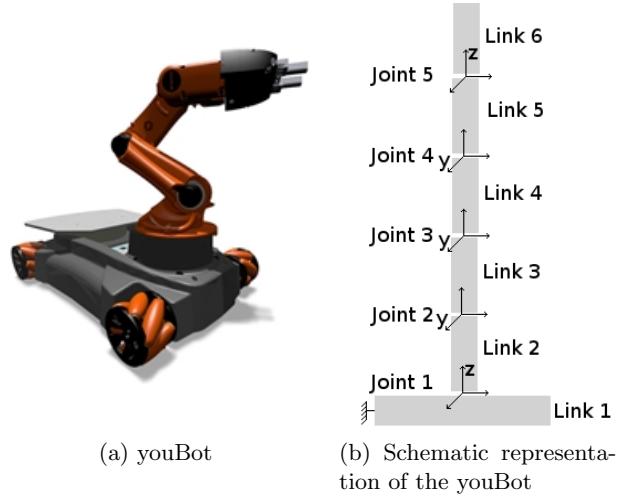


Fig. 1: The platform

The robotic manipulator has six links connected by 5 actuated rotational joints. The axis of rotation of joint 1 and 5 is the z-axis in the frame depicted, for joint 2,3 and 4 the axis of rotation is the y-axis.

The four Mecanum wheels are mounted to the first link of the robotic actuator to make it mobile, this is shown in Fig. 2.

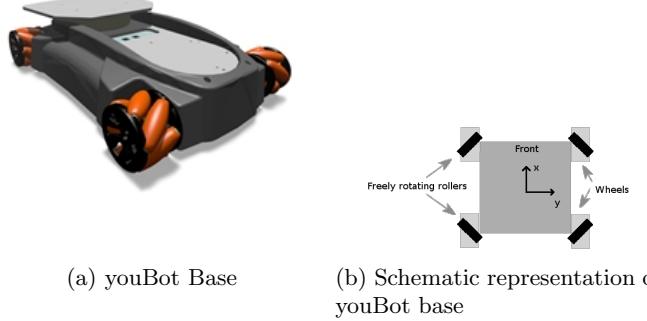


Fig. 2: The platform base with Mecanum wheels

3 Model of the platform

3.1 Submodel of a link

A modeling process begins with defining a dominant behavior of the component. For robotic applications, the dominant behavior of the links in the manipulator is the behavior of a body to which external wrenches are applied. As long as the youBot is used within the specifications, it is valid to assume that the links are rigid and are treated as rigid bodies. However, the submodel can be adapted to include other behavior without effecting the interface with other parts of the youBot model.

Modeling rigid body dynamics of a link The model of rigid-body dynamics consists of inertia and gyroscopic effects. According to screw theory the kinematics of a rigid-body motion can be represented as a rotation (ω) about an axis along with translation (v) along the same axis [1]. Furthermore, we will use the following notation:

$$- T = \begin{pmatrix} \omega \\ v \end{pmatrix}: \text{twist}$$

- $W = \begin{pmatrix} \tau \\ F \end{pmatrix}$: wrench
- $I = \begin{pmatrix} J & 0 \\ 0 & M \end{pmatrix}$: inertia tensor placed in the center of mass.
- F : force
- P : momentum screw
- τ : torque
- M : mass matrix
- J : inertia matrix

The wrench balance representing the inertia effects, expressed in the principal inertia frame k , (in the center of gravity and oriented along the three primary inertia axes) will have a form [8, 7]:

$$I^k \dot{T}_a^{k,0} = \begin{pmatrix} \tilde{P}_\omega^k & \tilde{P}_v^k \\ \tilde{P}_v^k & 0 \end{pmatrix} T_a^{k,0} + (W^k)^T \quad (1)$$

where

- k is the principal inertia frame of the body
- 0 is the inertial frame
- a is the body

In the above relation the component for an inertia can be recognized on the left hand side of the equal sign. The first component at the right hand side of the equal sign represents the fictitious forces and torques (wrenches) including the gyroscopic effect, the second component at the right hand side of the equal sign represents externally applied wrenches. A wrench balance as described by the formula above is represented in a bond graph by a 1-junction. The bond graph shown in Fig. 3 represents the equation above.

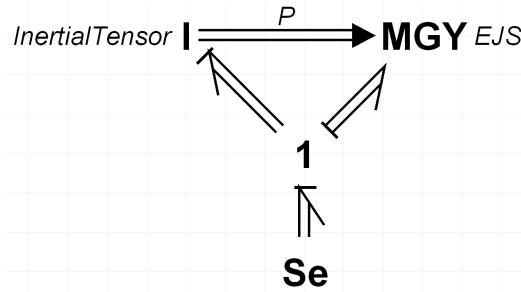


Fig. 3: A bond-graph model of a rigid body including gyroscopic effects

In this model the fictitious wrenches including the gyroscopic effects are expressed by a gyrator (MGY element). A gyrator represents a power continuous

relation between efforts and flows. The externally applied wrenches are represented by the Se (Source of effort) element.

Externally applied wrenches The externally applied wrenches include joint reactions, collisions, gravity etc. The most significant ones, that have a constant effect on the link dynamics, are the joints reactions and gravity. The joints will be connected to a link by means of power ports, through which the joints apply wrenches to the links. However, these wrenches are expressed in the frame that corresponds with the actuated joint. The wrench balance, as discussed in the previous section, is represented in a principal inertia frame of the body. Therefore, a transformation of coordinates is required between these parts of the model.

A homogeneous matrix H_i^j represents the position of a frame i with respect to a frame j with a translation component p and a rotation component R :

$$H_i^j = \begin{pmatrix} R_i^j & p_i^j \\ 0 & 1 \end{pmatrix} \quad (2)$$

A twist, in matrix form[7], is defined as the product of a position and the derivative with respect to time of the same position[7]:

$$\begin{aligned} \tilde{T}_i^{i,j} &= H_j^i \dot{H}_i^j \\ \tilde{T}_i^{j,j} &= \dot{H}_i^j H_j^i \end{aligned} \quad (3)$$

These two expressions can be combined into:

$$\tilde{T}_k^{j,l} = H_i^j \tilde{T}_k^{i,l} H_j^i \quad (4)$$

Which represents a change of coordinates for twists in matrix form. It is possible to see [6, 7] that a change of coordinates for twists in vector form, as applied in this paper, can be expressed as:

$$T_k^{j,l} = Ad_{H_i^j} T_k^{i,l} \quad (5)$$

with

$$Ad_{H_i^j} = \begin{pmatrix} R_i^j & 0 \\ p_i^j R_i^j & R_i^j \end{pmatrix} \quad (6)$$

the adjoint of the homogeneous matrix H_i^j .

Since a change of coordinates is power continuous, an expression for a change of coordinates for wrenches can be found:

$$W^j T_k^{j,l} = W^j Ad_{H_i^j} T_k^{i,l} = (Ad_{H_i^j}^T W^j)^T T_k^{i,l} = W^i T_k^{i,l} \quad (7)$$

such that

$$(W^i)^T = Ad_{H_i^j}^T (W^j)^T \quad (8)$$

Since the relations are power continuous, a TF element is used to represent this change of coordinates in the bond-graph language.

Coordinate transformations are required between three frames:

1. The body-fixed frame in the previous joint, from now on called frame i
2. The body-fixed frame in the next joint, from now on called frame j
3. The principal inertia frame of the link, from now on called frame k.

To relate these three frames only two changes of coordinates are required since the third change of coordinates can be composed of the other two. For example:

$$H_j^k = H_i^k H_j^i \quad (9)$$

In equation(9) the two changes of coordinates changes are applied between:

1. Frame i and frame k
2. Frame j and frame i

And frame i has a central role. The choice can be made to give either frame i , frame k or frame j this central role. This does not effect the model behavior or reusability of the model. In this paper, frame i has the central role. Therefore, the changes of coordinates are applied as shown in the example.

Since deformation of the link is neglected, it is assumed that the above mentioned changes of coordinates are constant in time. However, the model can easily be extended to include those effects by replacing the linear relation with non-linear, time-dependent ones. Fig. 4 shows the model of the link including the discussed changes of coordinates. p and p_1 are the power ports to which the joints will be connected.

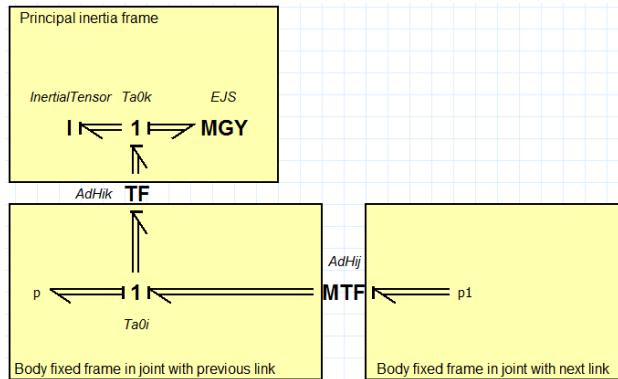


Fig. 4: A bond-graph model of a rigid body with changes of coordinates to the joints

Gravity, expressed in the inertial frame 0, is a constant force (wrench) that applies to the body in the negative z-direction of this frame and can be modelled as a Se element in the bond-graph language. The other parts of the model are

not expressed in the inertial frame and therefore a transformation of coordinates is required between the inertial frame and one of the other frames. Since the frame in the previous joint is used as a starting point, it is chosen to apply the coordinate transformation between system 0 and frame i. The H-matrix that corresponds to this change of coordinates is passed on by the previous joint as a signal that modulates the transformations.

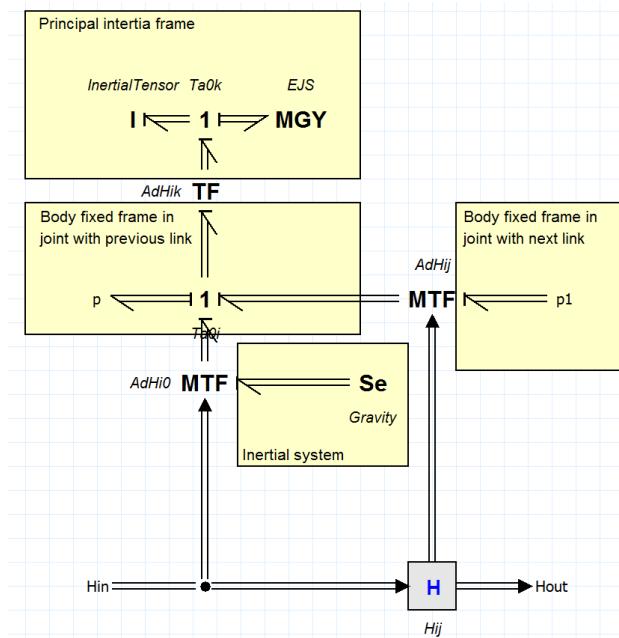


Fig. 5: A bond-graph model of a link

Fig. 5 shows the complete submodel of a link. In this model, the Se element representing gravity can be found with the coordinate transformation between the inertial frame 0 and frame i. The signal port Hin is used to obtain the position of frame i with respect to the inertial frame. This is then multiplied by H_i^j , the relative position of frame j with respect to frame i , to obtain the position of frame j with respect to the inertial frame. This position is passed on to the next joint through signal port Hout. The ports p and $p1$ are the power ports of the link, to which the joints will be connected.

3.2 Submodel of a joint

Just as with the submodel of a link, it has to be decided which behavior will be modelled and which behavior will be neglected. For this model, it is decided only to model the transfer of energy in a joint. Therefore, the following behavior has explicitly been neglected:

- Friction in the joint
- Energy storage in the joint

However, the model can easily be extended to include these and other effects.

Since a joint establishes an energetic connection between two links, it imposes a relation between the wrenches and the twists of the two bodies.

A wrench applied in a joint is applied between the two connected links and, in fact, applies the same wrench to both links (action and reaction). This specifies the relation between the wrench applied by the actuator and the wrenches applied to the connected links:

$$W_{act} = W_{link1} = W_{link2} \quad (10)$$

On the other hand, there is a relation between the twists of link 1 with respect to the inertial frame, expressed in frame n ($T_1^{n,0}$) and the twist of link 2 with respect to the inertial frame, expressed in frame n ($T_2^{n,0}$). Just as with other flow type variables the following relation holds:

$$T_2^{n,0} = T_1^{n,0} + T_2^{n,1} \quad (11)$$

The above two relations are elegantly and simply represented by a 0-junction in the bond-graph language. For these relations to apply, all twists and wrenches should be expressed in the same frame. For this paper frame i is selected, this does not effect reusability of the model. A joint is an energetic connection between two links. The twists and wrenches of these two links are expressed in frames fixed with the link. As a result they are expressed in different frames and a change of coordinates is required.

A joint performs two tasks in a system: It applies constraints in certain directions of motion while allowing freedom in other directions of motion. When no forces can be applied in the directions of freedom the joint is considered unactuated, when forces can be applied by means of actuators the joint is considered to be actuated. A joint can either be one-dimensional or multi-dimensional. It can either be translational, rotational or a combination of these two. For the robotic manipulator used as case study in this document all joints are actuated rotational joints with one degree of freedom. That is: motion around a specific geometric line is allowed while all other motions are constrained.

For a rotational joint a simple relation can be defined between the actuator torques and the wrench applied in the joint when the screw axis is placed in and aligned with the joint:

$$W = \hat{W}\tau \quad (12)$$

Where, in case of a rotational joint:

$$\hat{W} = \begin{pmatrix} \hat{\tau} \\ 0 \end{pmatrix} \quad (13)$$

and $\hat{\tau}$ is the axis around which the torque is applied. Due to power continuity this imposes a relation between the corresponding component of the twist in the joint and the joint velocity:

$$W^T T_a^{i,b} = (\hat{W}\tau)^T T_a^{i,b} = \tau \hat{W}^T T_a^{i,b} = \tau \dot{q} \quad (14)$$

such that

$$\dot{q} = \hat{W}^T T_a^{i,b} \quad (15)$$

Since this set of relations is power continuous, they can be represented by a transformer in the bond-graph language.

Since the joint actuator is related to one of the wrench components and the joint is constrained in the other directions motion constraints need to be added in the model. A constraint can be modelled as a high stiffness and a low compliance connected to the joint with effort out causality. Fig. 6 shows the model of a joint.

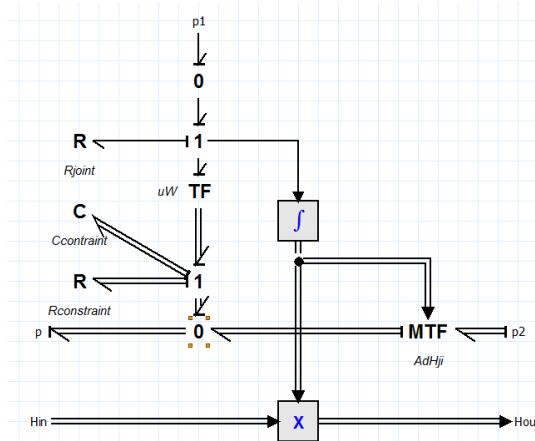


Fig. 6: A bond-graph model of a joint

The change of coordinates is represented by the MTF element "Ad*ji*". The conversion from joint torque to wrench and from twist to joint velocity is represented by the TF element "u*Tbai*".

In the bond-graph model two other blocks are present, an *f*-block and an *x*-block. The *f*-block calculates the joint position based on an integration of the joint velocity and uses this to construct the H matrix that represents the position

of the joint: H_j^i . This matrix is then used for the coordinate transformation. A multiplication block indicated by the symbol \times multiplies two matrices. The two H-matrices that are multiplied are:

- The position of frame i with respect to the inertial frame 0, obtained from $H_{in}^0 (H_i^0)$
- The position of frame j with respect to frame i (H_j^i)

such that the position of frame j with respect to the inertial frame is obtained:

$$H_j^0 = H_i^0 H_j^i \quad (16)$$

3.3 Submodel of a Mecanum wheel

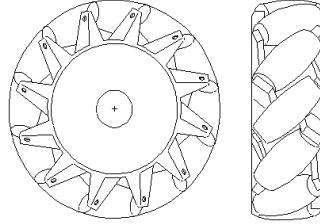


Fig. 7: An example of a Mecanum wheel (Source: www.robotika.sk)

A Mecanum wheel is a wheel with rollers mounted on it. These rollers are mounted in an angle of 45° with respect to the wheel and can freely rotate. Fig. 7 shows an example of a Mecanum wheel.

In this paper a Mecanum wheel is considered as a construction that enables the application of forces to the robot by actuation of the wheel axis. All dynamic behavior of the Mecanum wheel is neglected for simplicity reasons.

A Mecanum wheel is, first of all, a wheel. Therefore, the model of a wheel is first discussed.

Model of a wheel We considerer here the ideal kinematic relation of a wheel not modelling any visco-elastic behavior. This can be easily extended within this framework as shown in [9]. An ideal wheel on a surface, as shown in Fig. 8, creates a relation between the rotational velocity ($\dot{\phi}$) of the wheel and the translational velocity (v) of the wheel with respect to the surface. It is common knowledge that the radius (r) of the wheel is a measure for this relation. More specifically:

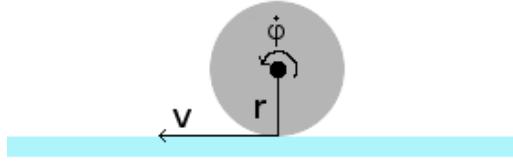


Fig. 8: A common wheel represented as transformer

$$v = r\dot{\phi} \quad (17)$$

On the other hand, a wheel also imposes a relation between the torque (τ) applied to the wheel and the force (F) between the wheel and the surface. The radius also is a measure for this relation:

$$F = (1/r)\tau \quad (18)$$

When comparing the energy flow, or power, on the two sides (power ports) of the wheel (translation and rotation) it is clear that it is equal:

$$P_{translation} = vF = r\dot{\phi}(1/r)\tau = \dot{\phi}\tau = P_{rotation} \quad (19)$$

In the bond-graph language this ideal behavior is described by the standard element transformer.

Extension of the model of a wheel to the model of a Mecanum wheel
At the contact with the surface, a Mecanum wheel has rollers where a common wheel has none. These rollers can rotate freely around one axis and are mounted in an angle of 45° with respect to the transformation direction of the wheel. Fig. 9 clarifies the position of a roller (grey, filled rectangle) in the wheel (black rectangle outline). In Fig. 9 two coordinate frames are shown:

1. One frame that is aligned with the wheel, this frame is called frame 1
2. A second frame that is aligned with the roller, this frame is called frame 2

In Fig. 9 the axis around which the wheel rotates is the x-axis of frame 1 and the axis around which the roller can rotate is the y-axis of frame 2.

Since the roller can rotate freely around its y-axis, it acts as a bearing for translational movements in its x-direction. Just like in a common bearing the friction is minimized, for this model it is assumed that no friction is present. As a result of the lack of friction in this direction, there is no relation between:

1. the force applied by the wheel/ velocity of the wheel and

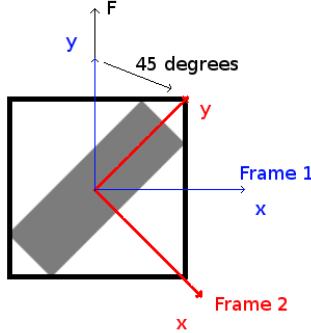


Fig. 9: Forces in frame 1 aligned with the wheel (blue) and frame 2 aligned with the roller (red)

2. the force applied between the wheel and the surface/ velocity of the wheel with respect to the surface

in the x-direction of the roller. Any force applied in this direction will only contribute to the rotation of the wheel. When putting this in perspective with the model of the wheel discussed earlier, there is a transformation ratio of 0 in this x direction of frame 2. Note that this is expressed in a different frame than the transformation of the wheel, which is expressed in frame 1.

To model the effect of the rollers in the bond-graph language, a 2-dimensional transformer element is used to represent the effect of the roller in frame 2. The transformation ratio (r) of this transformer is:

$$r = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (20)$$

It has a transformation ratio of 0 in the x direction and a transformation ratio of 1 in the y direction. Note that this transformer represents a relation between two translational components.

Since the transformation of the wheel is expressed in frame 1 and the transformation of the roller in frame 2, one of the two should be expressed in the frame of the other. Such that, both transformations are expressed in the same frame. It is chosen to express the transformation of the roller in frame 1.

To express a transformer in a different frame, the transformation ratio should be premultiplied by the change of coordinate matrix A and post-multiplied by its inverse:

$$TF^j = A_i^j T F^i (A_i^j)^{-1}, \text{ where } (A_i^j)^{-1} = A_j^i. \quad (21)$$

In the bond-graph language, this change of coordinates can be represented by transformers as shown in Fig. 10.

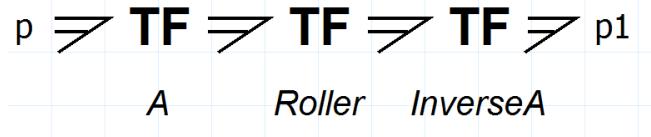


Fig. 10: Change of coordinates for a transformer in the bond-graph language

In case of a pure rotation [7]:

$$A = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \quad (22)$$

Where ϕ is the rotation between frame i and frame j .

When going from frame 2 to frame 1 there is a rotation of -45° , so in this case ϕ equals -45° .

Now the bond-graph model for the complete Mecanum wheel can be constructed as shown in Fig. 11.

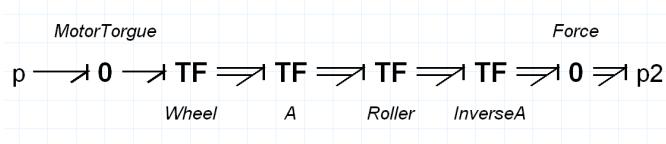


Fig. 11: The complete bond-graph model of a Mecanum wheel

Please note that in practise two types of Mecanum wheels are used:

1. One where the rollers are mounted in an angle of 45° with respect to the wheel and
2. a second type where the rollers are mounted in an angle of -45° with respect to the wheel

Figure 2b illustrates this. The difference results in slightly different models. However, both models are structured as explained in this section.

3.4 Construction of the model

The model of the robotic manipulator is constructed as a serial link in an object oriented way. In this model, two of the previously discussed submodel types are used:

1. submodels of links, connecting two joints
2. submodels of joints, connecting two links

such that the model of a robotic arm is as shown in Fig. 12.

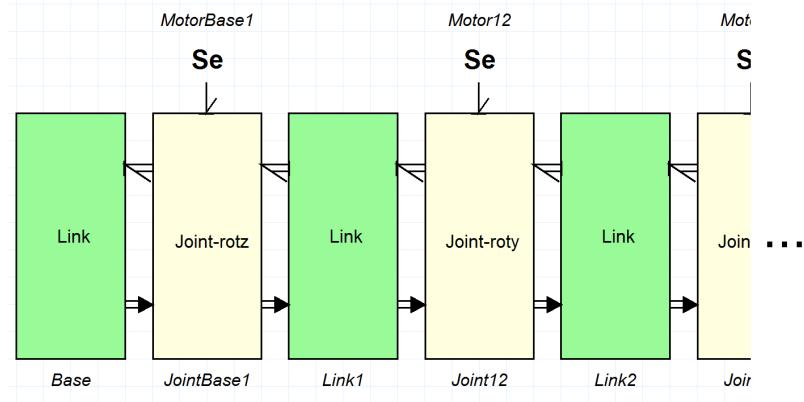


Fig. 12: Model of a robotic arm

The Se elements connected to all the joints represent the torques applied by the actuators in the joints. If necessary, the model can be extended with submodels of the electric motors. Furthermore, two connections can be seen between all joints and links:

1. a power connection establishing energetic coupling of the elements
2. a signal connection that passes on the homogeneous matrix representing the position of the connection point

So, every link and joint passes the position of the connection point with the next element on to the next element.

Connection of the Mecanum wheels to the robotic manipulator As described in section 2, four Mecanum wheels are connected to the first link of the robotic manipulator. However, in the model they cannot be connected directly, since the model of the robotic manipulator is using 6 dimensional vectors using screws and the wheels to not. The model of the Mecanum wheels provides a linear force caused by the Mecanum wheel as a result of the axis actuation. This linear force is applied in a point of the link. If this would be transformed to a wrench, the wrench would be applied to the body. This is a clear advantage in addition to the earlier mentioned advantage of being able to address rotations and translations simultaneously.

Transformation from a force in a point to wrench applied to a body and from a twist of a body to a velocity of a point is power continuous: the operation does not add energy to or remove energy from the system. Transformation of a twist to a linear velocity in a point (p) is given as:

$$\dot{p} = \omega \wedge p + v \quad (23)$$

Where a twist has the following form:

$$T = \begin{pmatrix} \omega \\ v \end{pmatrix} \quad (24)$$

This relation can be expressed in matrix form:

$$\dot{p} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} 0 & z & -y & 1 & 0 & 0 \\ -z & 0 & x & 0 & 1 & 0 \\ y & -x & 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \omega \\ v \end{pmatrix} \quad (25)$$

Based on power continuity, the relation between a force and the corresponding wrench can be found:

$$F^T v = F^T A T = W^T T \text{ so } F^T A = W^T$$

when transposed on both sides :

$$A^T F = W$$

These two relations together are the relations of a transformer in the bond-graph language, as shown in section 3.3. So, a transformation from screws to linear velocities/ forces and vice-versa can be accomplished with a multi-dimensional transformer in the bond-graph language. Please note that the transformation transforms between a three dimensional force/ velocity and a wrench/ twist. The Mecanum wheels exert no force in the z-direction and therefore the force vector is extended to:

$$F = \begin{pmatrix} F_x \\ F_y \\ 0 \end{pmatrix} \quad (26)$$

With all the forces applied by the wheels transformed to wrenches applied to the link, they can be summed. In the bond-graph language, this is represented by a 1-junction. The summed wrench can then be applied to the base. Fig. 13 shows the complete submodel of the Mecanum wheels. The blocks "Mecanum wheel type A" and "Mecanum wheel type B" are submodels containing the model of a Mecanum wheel as discussed earlier. The Se elements are sources of effort, they apply a fixed torque to the axis of the Mecanum wheels.

Since the position of the base is required in the base block, an integrator-block (\int) has been added. This block calculates the position of the block based on the current position and the twist of the base. But this should be done with care and integrating in a specific way, following the relation reported in equation 3. Furthermore, an R element (resistance) and a C element (stiffness) can be found in the model. These elements only act in the z-direction of the base and represent

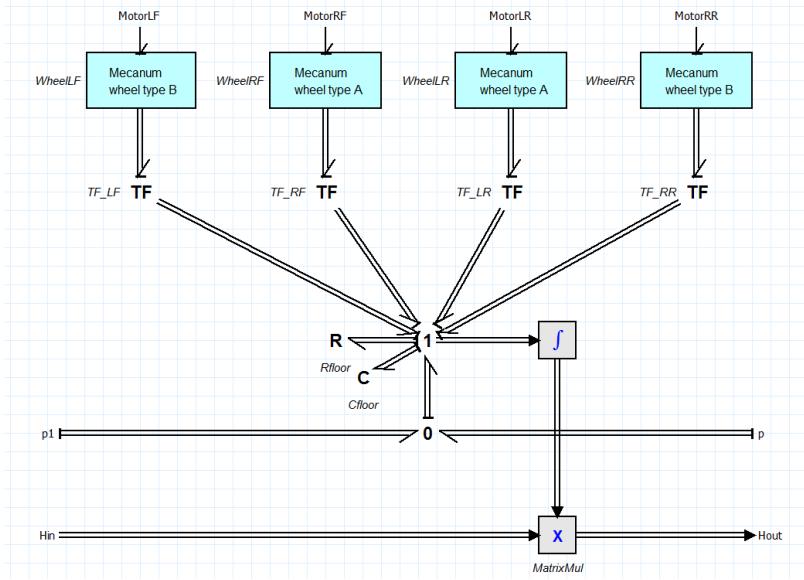


Fig. 13: Model of the Mecanum wheels

the contact with the floor. Since the floor is very stiff and has a high resistance, the values of these two components are high. Without these elements, the robot-model would accelerate in the negative z-direction due to gravity. Note that these elements do not represent the visco-elastic traction. The submodel can easily be connected to the first link of the robotic manipulator as shown in figure 14.

4 Conclusion

In this paper, a walk-trough tutorial on the modeling of a complex robotic system is presented. By writing the model in the bond-graph language, a model with reusable components has been created for an exemplary platform: youBot. In the paper it has been noted that the model, or submodels used in the model, can easily be adapted or extended with other behavior.

Validation of the model is not present in this paper since official validation still needs to be done with the actual platform. The platform was not available at the time of writing.

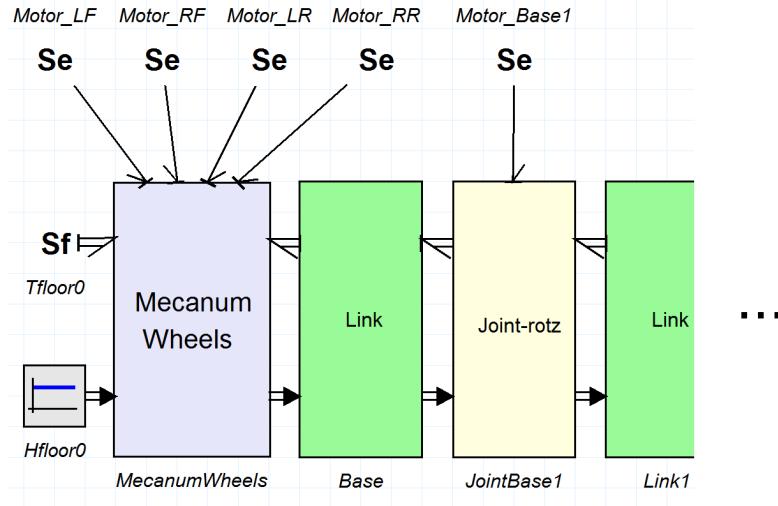


Fig. 14: Model of the Mecanum wheels connected to the first link

References

1. R. Ball. *A Treatise on the theory of Screws*. CAMBRIDGE: AT THE UNIVERSITY PRESS, 1900.
2. P.C. Breedveld. Modeling and simulation of dynamic systems using bond graphs. In *In Control Systems, Robotics and Automation, from Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO. EOLSS Publishers*, pages 1–36.
3. P.C. Breedveld. Port-based modeling of mechatronic systems. *Mathematics and Computers in Simulation*, 66(2-3):99 – 128, 2004. Selected papers from the 4th IMACS Symposium on Mathematical Modelling.
4. KUKA. Kuka youbo. url: <http://www.kuka-youbot.com/>, October 2010.
5. Locomotec. Kuka youbot store. url: <http://www.youbot-store.com/>, October 2010.
6. J.M. Selig. *Geometrical methods in robotics, Monographs in Computer Sciences*. Springer Verlag, New York, first edition, 1996.
7. S. Stramigioli and H. Bruyninckx. Geometry and screw theory for robotics. In *Proceedings IEEE International Conference on Robotics and Automation*, 2001.
8. S. Stramigioli and H. Bruyninckx. Geometry of dynamic and higher-order kinematic screws. In *Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea*, pages 3344–3349, 2001.
9. Stefano Stramigioli and Vincent Duindam. Port based modeling of spatio-temporal visco-elastic contacts, 2004.

3 Fault tolerant control

Fault tolerant control of mobile manipulators

Douwe Dresscher and Yury Brodskiy

University of Twente.

Abstract. Actuation failures in mobile manipulators present a considerable threat to the safety of the environment and the robot itself. To confront this problem, fault tolerant control should be employed. Herein an approach of increasing the resilience of interaction control to actuation and sensing failures is presented. The proposed solution includes a.)a model-based fault detection and isolation algorithm to obtain fault signalling and b.)behavioral reactive control that enables the manipulator to continue execution of the last command.

1 Introduction

For a robotic manipulator the typical failures that occur at loop level control jeopardize the safety of the robot. With a position control a robot will react on such events with immediate acceleration towards unsafe velocities and forces. This is a problem for mobile robots since there are no safety barriers in an open environment. Therefore, an important feature of an autonomous robot is to detect when its normal work-flow is interrupted, to identify the cause of the interruption and to respond such that the robot is able to complete its task if possible. A well known approach to detect and identify a failure is the use of Fault Detection and Isolation (FDI) algorithms as discussed in Venkatasubramanian et al. (2003b,a,c). In addition to the development of a model based FDI algorithm this paper treats the development of an adaptable controller. Kinematic redundancy(as defined in Conkur and Buckingham (1997)) in the mobile manipulator is used to get its end-tip in an arbitrary position within its workspace when an actuator or sensor failure makes a joint or Mecanum wheel unusable.

Section 2 describes the platform and control algorithm used in the paper. Section 3 treats the development of a bond-graph-observer based FDI algorithm. Section 4 discussed the reconfigurations applied to the controller and plant to enable it to complete its task in the presence of failures. Section 5 presents the simulation results.

2 Platform and control algorithm

2.1 Description of the platform

For this paper, the youBotLocomotec (2010); KUKA (2010) is used as an example. The kinematic structure of the youBot is shown in Fig. 1b. The robotic manipulator has six links connected by 5 actuated rotational joints. The axis of rotation of joint 1 and 5 is the z-axis in the frame depicted, for joint 2, 3 and 4 the axis of rotation is the y-axis. Four Mecanum wheels are mounted to the first link(base) of the robotic actuator to make it mobile. This is shown in Fig. 2. A detailed platform description and a dynamic model are presented in Dresscher et al. (2010).

2.2 Description of the Control algorithm

The controller applied for the control of the platform is an interaction controller as described in Stramigioli and Bruyninckx (2001); Stramigioli (2001). An interaction controller controls the dynamic relation between

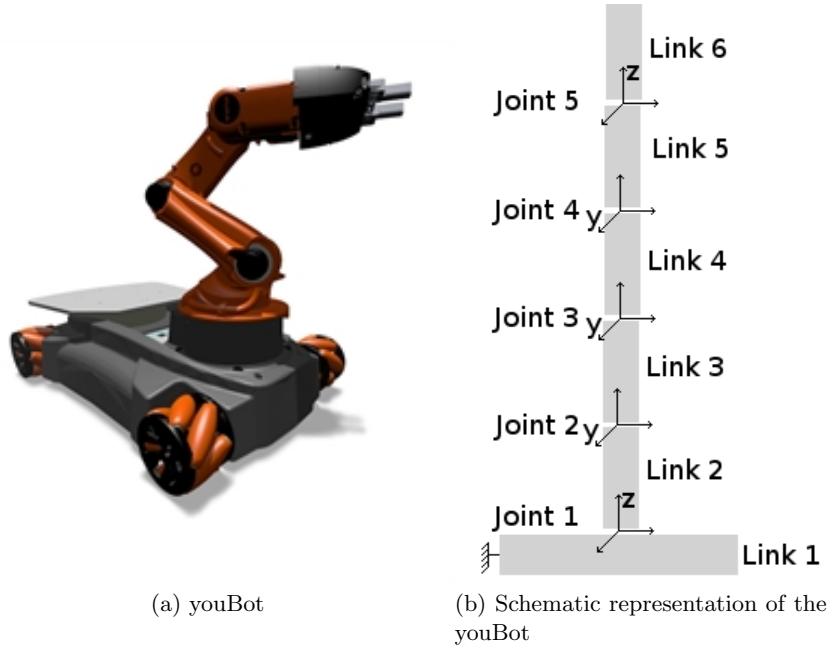


Fig. 1: The platform

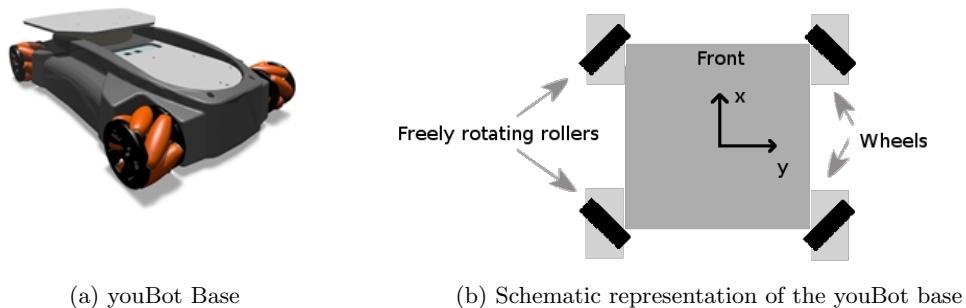


Fig. 2: The platform base with Mecanum wheels

conjugate force (F) and velocities(dx/dt)/ displacements(x). Where the displacement(x) is the difference between the actual and the desired position. For example:

$$F = Kx + R(dx/dt) \quad (1)$$

In 1 the actuation force is dependent on 2 dynamic relations, K and R , and 2 variables, x and dx/dt . Instead of controlling the position (x) or the force (F) the dynamic relations (K and R) can be changed. A combination of spring, mass and damper elements is used to create a stable controller. This way of controlling has several advantages Stramigioli and Bruyninckx (2001); Colgate and Hogan (1989, 1988, 1987):

1. Passive elements such as springs and dampers will never store more energy than the energy that is stored at $t=0$ plus the energy supplied by the environment. The same holds for a network consisting of such passive elements. Therefore, stability of such a network is guaranteed.
2. Naturally robust controller
3. Compliant behavior of the robot if the spring elements are dimensioned properly
4. Intuition in the control design

It can be applied when the following conditions are satisfied:

1. The plant should be back-drivable
2. The friction in the system should be low
3. Position measurements should be available

The controller can intuitively be seen as a spatial spring applied between the tip of the robotic manipulator and a virtual position. The robustness of this controller can be explained as follows. A virtual spring element will exert forces at its connection points. The force in one connection point will be directed toward the other connection and the combination of forces will act until the system is in equilibrium position: $\Delta x = 0$. If the equilibrium point cannot be reached the connected point will go to some minimal Δx . This minimal Δx is not necessarily the global minimum. In order to overcome this limitation a supervisory planner should be added to the system, which is outside the scope of this paper. A damper element is working the same way, with the only difference that it will apply forces to get $\Delta v = 0$. As long as a robotic manipulator has sufficient degrees of freedom this reasoning holds.

In the interaction controller used in this design, a spatial spring component is applied in the work-space of the manipulator. The actuation and sensing is done on the joints and wheels (called actuator space). The transformations between the work-space and the actuator-space are done in two steps:

1. A Geometric Jacobian is used to transform between the work-space and a joint-space. In this joint space the base with Mecanum wheels is represented by a joint with the three degrees of freedom of the base: translation along the x-axis of the base, translation along the y-axis of the base and rotation around the z-axis of the base. The joint-space is eight-dimensional, five belong to the joints and three to the base.
2. The Moore-Penrose pseudo-inverse of a Jacobian based on the relative positions of the Mecanum wheels is used to transform between the joint-space and the actuator-space. The actuator-space is nine-dimensional, five belong to the joints and four belong to the wheels.

The above two mappings are power-continuous and therefore the passive character of the controller is maintained.

3 Fault Detection and Isolation

The development of a detection system begins with the systematic analysis of possible abnormal events in the system. The systematic analysis for manipulator failures was done by Visinsky and Visinsky (1991). Comparing this result with the statistical study done by Carlson (2004) we can infer that there are two types of failures that should be considered at loop-level control: sensing and actuation failures. Actuation failures are most likely to originate from joints or wheels. The links have negligible probability of destruction with a prescribed workload whereas the MTBF for a joint is in the range of 5000 hours (Groom et al. (1999)).

3.1 Motivation for detection method

In Venkatasubramanian et al. (2003b,a,c) a systematic and comparative study on detection methods is presented from different perspectives. In the work the FDI algorithms are divided in three categories based on the knowledge utilized in the algorithm:

1. Quantitative model-based methods
2. Qualitative model-based methods
3. Process history based methods

An important conclusion of their work is that none of the mentioned FDI methods satisfies all the desirable characteristics of an FDI algorithm and that a hybrid solution should be found. A hybrid solution is a combination of model-based and history-based methods. The development of a hybrid solution starts with the development of a model-based FDI, which is further enhanced with history based adaptation. Inhere the first step toward such a solution is presented: the development of an observer-based FDI. The development of a hybrid method is subject to future work. The work described in Dresscher et al. (2010) provides a re-usable bond-graph model. These resources can be used to create an observer-based (Quantitative model-based) FDI. The bond-graph model can be re-used as a non-linear observer model, thus avoiding one of the primary concerns of Venkatasubramanian et al. (2003a) on observer-based FDI's: that they would only be applicable to linear models.

3.2 Architecture of the observer based FDI

Presuming that the sensors in a joint is limited to a position sensor the isolability is limited by those measurements. Figure 3 shows the observer-plant architecture used for the realisation of the observer-based FDI algorithm and is based on the position measurements that are available. The architecture can be described as a ghost of the actual manipulator connected to the manipulator and interaction controller in the actuation-space. The interaction controller consists of spring(C_{oc}) and damper(R_{oc}) elements. Such architecture employs two central ideas proposed by Polderman and Willems (1998):

1. The observer contains a competent model of the plant.
2. The observer is driven by *innovations*, by the error feedback, that is, by a signal that expresses how far the actual observed output differs from what we would have expected to observe.

For the damper elements in the interaction controller the velocity difference between the plant and the observer is required. The joint/wheel velocities of the observer are known, but the velocities of the plant cannot be measured directly. To obtain an estimate of the plant velocities interconnected control is applied as described in Stramigioli (2001). In figure 3 the interconnection consists of the elements C_i and I_i : a spring-mass combination is placed in series with the control spring-damper as shown in figure 4. When $I_i \ll I_p$ and $C_i \ll C_{oc}$ the velocity and position of I_i are a good estimate of the velocity and position of the plant while for C_i only the *position* of the plant is required.

3.3 Detection and isolation

The inconsistencies between the actual and the expected behavior(called residuals), should be such that actuator and sensor failures in the joints can be detected and a failing effector can be isolated. The residuals available based on the position sensor and the FDI algorithm are the joint/wheel position differences between the observer and the plant. This section discusses how these residuals can be used to detect sensing and actuation errors. Furthermore the level of isolation that is possible with the algorithm is discussed.

Causal analysis of the observer-plant (as described in Mukherjee (2006)) uses causal paths to prove detectability and isolability of faults. First the causal path analysis of the robotic arm is done, followed by the base with Mecanum wheels. Figure 5 shows one of the causal paths in the observer-plant system of the robotic arm. Overall the following causal paths exist between the known actuator inputs and the measured sensor values:

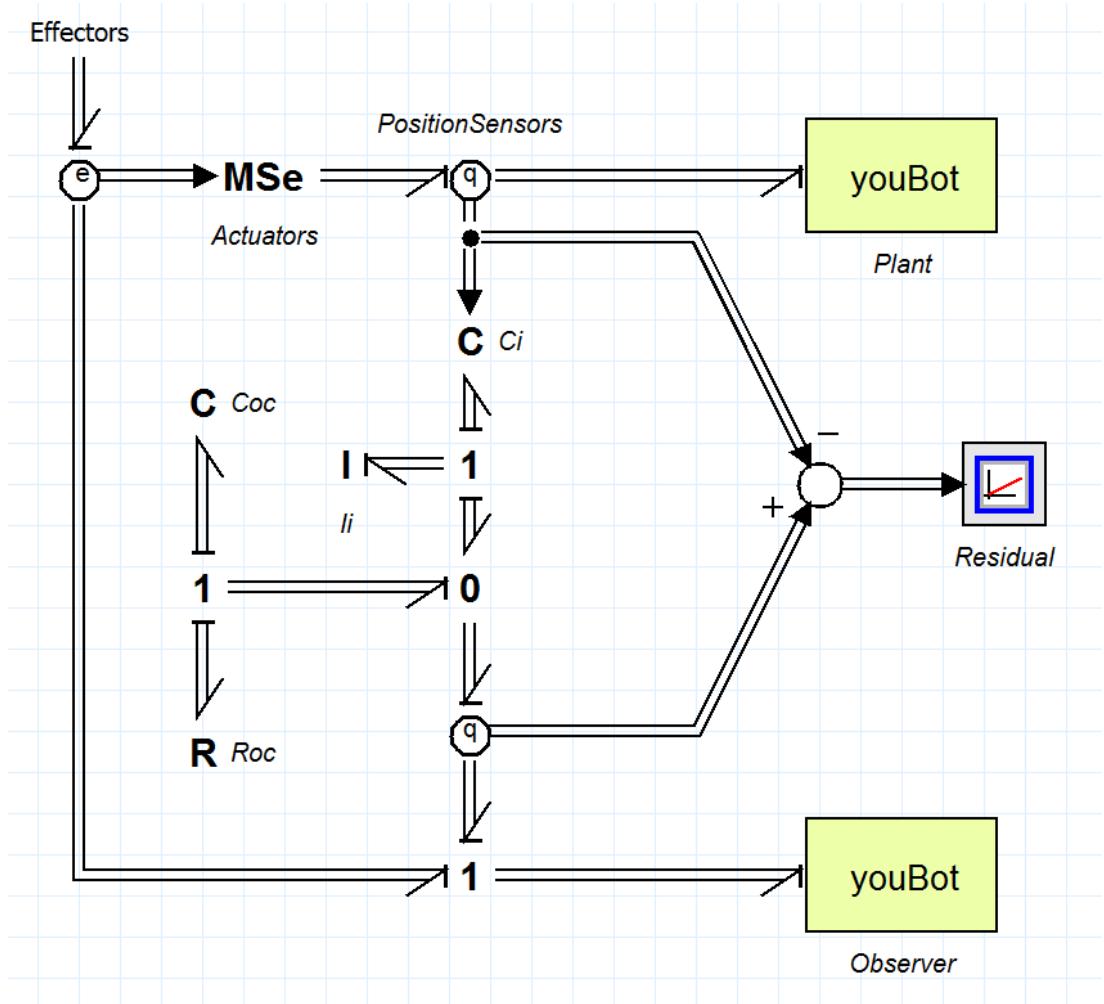


Fig. 3: Observer-Plant construction in bond-graph notation

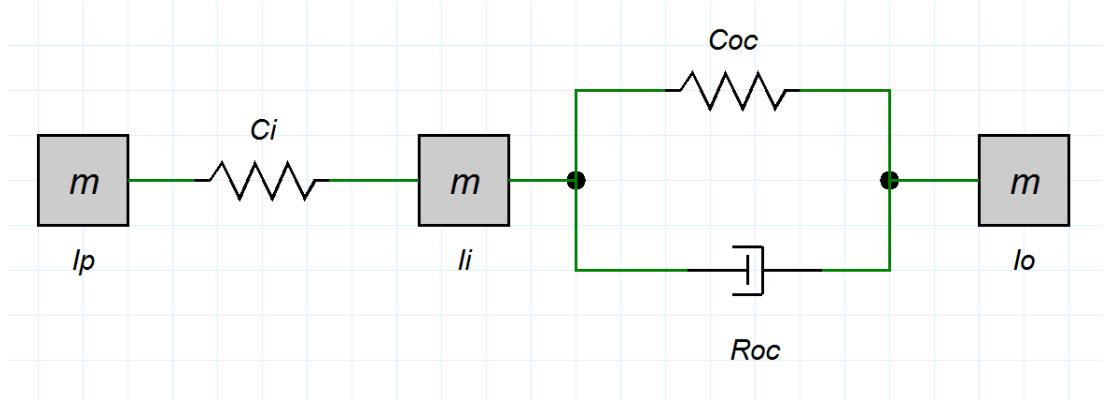


Fig. 4: Interconnected control, iconic diagram representation

1. $\text{Effectors} \rightarrow \text{Actuators} \rightarrow \text{Plant} \rightarrow \text{PositionSensors} \rightarrow \text{Residual}$
2. $\text{Effectors} \rightarrow \text{Actuators} \rightarrow \text{Plant} \rightarrow \text{PositionSensors} \rightarrow C_i \rightarrow I_i \rightarrow C_{oc} \rightarrow \text{Observer} \rightarrow \text{Residual}$
3. $\text{Effectors} \rightarrow \text{Actuators} \rightarrow \text{Plant} \rightarrow \text{PositionSensors} \rightarrow C_i \rightarrow I_i \rightarrow R_{oc} \rightarrow \text{Observer} \rightarrow \text{Residual}$
4. $\text{Effectors} \rightarrow \text{Observer} \rightarrow \text{Residual}$

From which the following signature matrix can be derived for the actuators and sensors:

	Residual
Actuators	1
PositionSensors	1

This shows that both actuation errors and sensing errors are detectable but not isolable. The detection system will have "effector failure" as base event, this is the highest isolation possible without adding any additional components and the residual space is equal to the feature space. A threshold can now be applied as mapping to the decision space. For the general purpose of the manipulator the reaction on such base event will provide sufficient increase of the safety. It will prevent wild motion of the manipulator in case of hardware failures. Please note that the load denoted as plant is dynamic but equal for plant and observer.

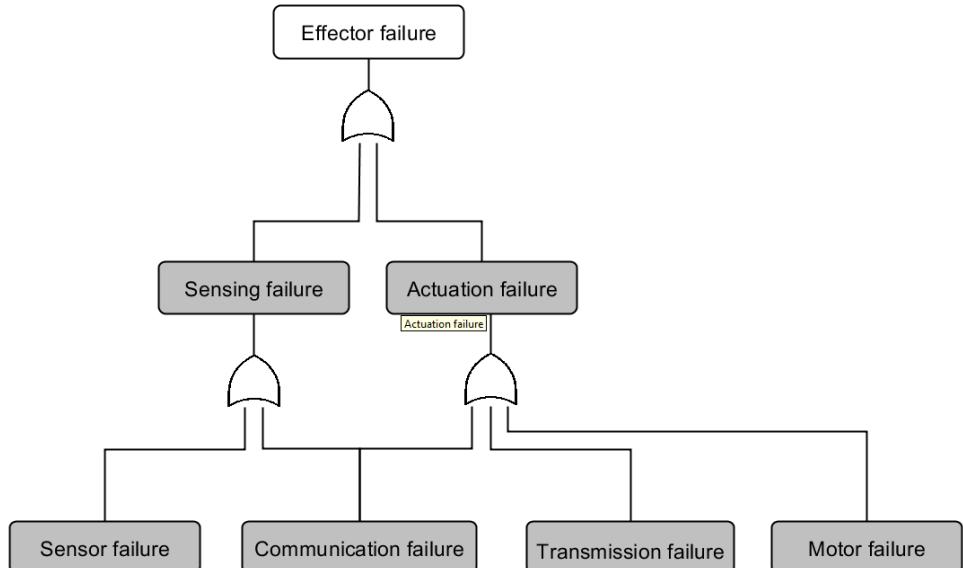


Fig. 5: FTA of an actuator

When multiple forces are applied to a single body such as the youBot base the forces are summed. The summed force is applied to the base with some dynamics D , and a velocity results. Based on this *single* velocity of the base, the velocity of the points of actuation can be computed using geometrical information on the relative actuator positions. Since all velocities result from the single velocity of the base faults are not directly isolable. We will prove that in case of a single failing wheel actuation the velocity of this wheel will have the residual that is largest in magnitude. This enables a mapping from the residual space to the feature space by determining the largest residual.

The wrench(W) applied to the body as a result of a weighted sum of forces(F) can be formulated as:

$$W = JF \quad (2)$$

Where J is some Jacobian matrix containing information on the relative actuation positions. Since the points of actuation cannot move with respect to each other, J is a constant matrix.

Using this a similar relation between the generalized momentum P and actuation momenta p can be found.

$$P = \dot{W} = \dot{J}F = \dot{J}F + J\dot{F} = Jp \quad (3)$$

Power continuity tells us that there should be a relation similar to equation 2 between the velocities of the actuation points(\dot{q}) and the twist(T) of the body:

$$\dot{q} = J^T T \quad (4)$$

While the twist of a body is, as mentioned above, related to the wrench applied to the body trough some dynamics D:

$$T = DW \quad (5)$$

For the robotic manipulator used in this case the dynamics of the base are dominant over the dynamics of the arm, when concerning the load of the Mecanum wheels, and can be simplified to a mass. Rewriting the above equation for the generalized momentum P:

$$T = I^{-1} P \quad (6)$$

Where I is the inertial tensor of the base in the center of mass. It is an invertible, diagonal matrix.

Combining the above equations gives a relation between the actuation velocities and the actuation momenta:

$$\dot{q} = J^T I^{-1} J p \quad (7)$$

Expansion shows that the matrix $J^T I^{-1} J$ has the following shape in this case:

$$J^T I^{-1} J = \begin{pmatrix} a+b+c & -a-b+c & a-b+c & -a+b+c \\ -a-b+c & a+b+c & -a+b+c & a-b+c \\ a-b+c & -a+b+c & a+b+c & -a-b+c \\ -a+b+c & a-b+c & -a-b+c & a+b+c \end{pmatrix} \quad (8)$$

where

- $a = I_{33}^{-1}(x+y)^2$
- $b = I_{44}^{-1}$
- $c = I_{55}^{-1}$

Due to the signs of the components, the diagonal terms are the largest in magnitude. A moment in one of the actuation points will cause a velocity for all four points, however the velocity of the point of actuation is the largest in magnitude. By detecting the residual with the largest magnitude a mapping to the feature space is created. A threshold function can now be applied to create the mapping from this feature space to a decision space.

4 Reconfiguration of the controlled manipulator

The next and final step in improving the resilience of the system to effector failures is to design a supervisory action on a failure such that the robot can complete its tasks if possible. Since the robot has kinematic redundancy it may be possible to complete its task with a broken wheel or joint. However, reconfiguration of the controlled manipulator is necessary to enable this.

4.1 Joints

When an actuation failure is present, the joint may reach an undesired state in which the workspace is unacceptable. For example, the joint may keep moving until it reaches an end-stop. When a position sensor fails the position of the tip is no longer known and control will be impossible. Both problems can be solved if the joint can be made a rigid connection at the point of failure since the joint position will then be fixed to the last known position. This can be achieved by equipping the robot with additional actuators, in the form of mechanical emergency breaks, on all the joints. With this addition a fault-tolerant actuation system (Isermann (2006)) is created. The breaks are cold stand-by actuators that will only be applied when a joint failure is detected.

It is desirable to stop actuating a failing joint to prevent further damage and to not use the position sensor of this joint since it may be unreliable. Therefore, the sensor readings should be removed from the control loop and control signals should be set to zero. This can be achieved by adapting the controller by making the column of the Geometric Jacobian corresponding to the failing joint zero.

4.2 Mecanum Wheels

The wrench applied to the body is the weighted sum of the forces applied by the Mecanum wheels. Using the Moore-Penrose pseudo-inverse of the Jacobian a combination of forces is selected that sum to the desired wrench. If one of the wheels cannot be actuated any more this solution is not valid any more. Therefore the pseudo-inverse should be adapted such that the solution fits the current configuration of the manipulator. This will automatically prevent the failing wheel from being actuated.

To minimize the effect of a failing wheel the torque contribution of this wheel should be minimized. Therefore, the resistance should be minimized. Since it is physically not feasible to reduce the resistance of the wheel the wheel should be left as it is.

5 Simulation results

Figure 6 shows the simulation results for the detection and isolation of faults. In these simulations a failure is injected at $t=0.1$. From this figure it can clearly be seen that the residuals respond to an error and that the joint or wheel is marked as failing.

Figure 7 shows the trajectory of the end tip of the robot from a starting position to a virtual position. The figure shows three trajectories:

- The trajectory in absence of a failure (green)
- The trajectory in the presence of an uncorrected failure (red)
- The trajectory in the presence of a corrected failure (Yellow)

It can be seen that the robot is unable to reach the desired virtual position although the interaction controller makes it go as close as the state of the robot allows. It is also clear that with the FDI algorithm and the response as described in this paper there is minimum deviation from the trajectory without faults.

6 Conclusions and future work

In this paper an observer-based FDI algorithm is discussed together with its actions in reaction to failures. Simulation results show that the algorithm is successful in increasing the robustness of a robot for the following failures:

- Joint actuation failures
- Joint sensing failures
- Wheel actuation failures
- Wheel sensing failures

As soon as the youBot is available for testing, the algorithm will be tested in practise.

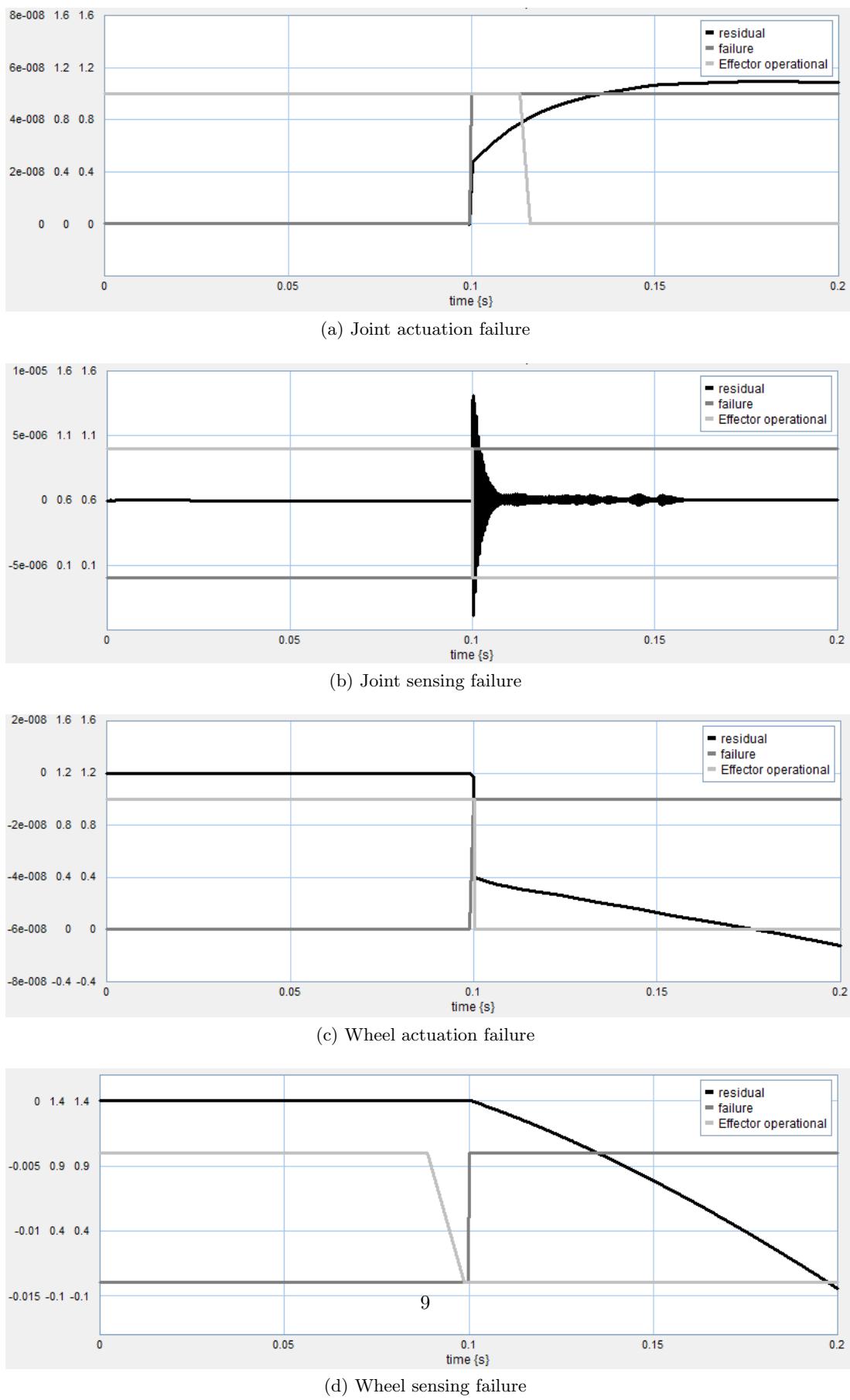
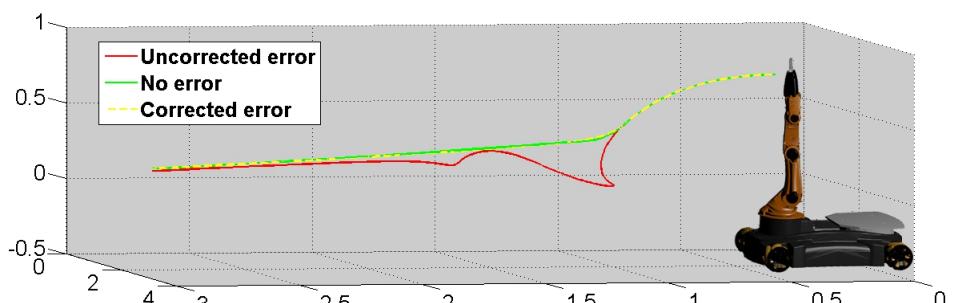
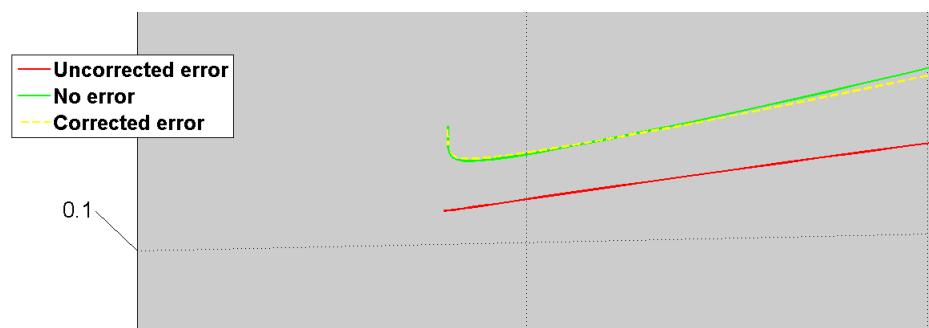


Fig. 6: Simulation results showing the detection and isolation of faults



(a) Trajectories



(b) End position error (zoomed in)

Fig. 7: Trajectories of the youBot moving to a virtual position

Bibliography

- Carlson, J. (2004). *Analysis of How Mobile Robots Fail in the Field*. PhD thesis, Citeseer.
- Colgate, J. and Hogan, N. (1987). Robust control of manipulator interactive behavior. *Modeling and Control of Robotic Manipulators and Manu*, 758.
- Colgate, J. and Hogan, N. (1988). Robust control of dynamically interacting systems. *International Journal of Control*, 48(1):65–88.
- Colgate, J. and Hogan, N. (1989). The interaction of robots with passive environments: Application to force feedback control. In *Fourth International Conference on Advanced Robotics*.
- Conkur, E. S. and Buckingham, R. (1997). Clarifying the definition of redundancy as used in robotics. *Robotica*, 15:583–586.
- Dresscher, D., Brodskiy, Y., Breedveld, P., Broenink, J., and Stramigioli, S. (2010). Modeling of the youBot in a serial link structure using twists and wrenches in a bond graph.
- Groom, K., Maciejewski, A., and Balakrishnan, V. (1999). Real-time failure-tolerant control of kinematically redundant manipulators. *Robotics and Automation, IEEE Transactions on*, 15(6):1109–1115.
- Isermann, R. (2006). *Fault-Diagnosis Systems*. Springer-Verlag, Berlin/Heidelberg.
- KUKA (2010). Kuka youbo. url: <http://www.kuka-youbot.com/>.
- Locomotec (2010). Kuka youbot store. url: <http://www.youbot-store.com/>.
- Mukherjee, A. (2006). *Bond graph in modeling, simulation and fault identification*. CRC Press.
- Polderman, J. W. and Willems, J. C. (1998). *Introduction to Mathematical Systems Theory - A Behavioral Approach*. Springer.
- Stramigioli, S. (2001). *Modeling and IPC control of interactive mechanical systems A coordinate-free approach*. Springer.
- Stramigioli, S. and Bruyninckx, H. (2001). Geometry and screw theory for robotics. In *Proceedings IEEE International Conference on Robotics and Automation*.
- Venkatasubramanian, V., Rengaswamy, R., and Kavuri, S. N. (2003a). A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. *Computers Chemical Engineering*, 27(3):313 – 326.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K. (2003b). A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers Chemical Engineering*, 27(3):293 – 311.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K. (2003c). A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers Chemical Engineering*, 27(3):327 – 346.
- Visinsky, M. and Visinsky, M. L. (1991). Fault detection and fault tolerance methods for robotics. Technical report.