Control of Brushless DC Motor with Static Redundancy for Force Feedback in Steer-by-Wire Applications

Alessandro Bertacchini, Luca Tamagnini, Matteo Mistrorigo, Paolo Pavan,

– University of Modena and Reggio Emilia –

In this paper we propose a Hardware-in-the-Loop (HIL) approach to implement a preliminary architecture for force-feedback control in steer-by-wire (SBW) applications. A brushless DC motor (BLDC) is used as force feedback actuator. The determination of the position of the BLDC rotor plays a key role in the control algorithm.

To obtain a reliable rotor position a classic triple static redundancy (TMR) is implemented. The position signals from the encoder integrated on the motor are computed in three different ways: using a 8-bit microcontroller, a 16-bit microcontroller, and, last, using the software module integrated in a virtual hardware development tool. The virtual hardware platform operates as voter, too. The position is the output of the voting algorithm and it is sent to the 16-bit platform that controls the motor and provides the correct output PWM signals.

The communication between virtual hardware and real hardware uses CAN bus. The bus is monitored by a dedicated development tool. Steer-by-wire is a safety critical application and therefore requires time-triggered protocols. In this preliminary architecture a dedicated network has been implemented and therefore the disadvantages of the event-triggered protocol are considerably reduced.

Experiments at different baudrates confirm that the voting algorithm produces correct results also in case of failure in one of the modules of the TMR architecture and it is not conditioned by bus loads. This means that the torque control algorithm of the BLDC motor can generate on the steering wheel (directly connected to the motor) a drive feeling like the one produced by a traditional steering system also in this fail-mode.

Introduction

What is a Steer-By-Wire system? SBW indicates a driver assistance system that requires a direct electronic control of the steer and replaces the traditional mechanical or hydraulic back-up by distributed fault-tolerant systems.

A complete SBW system gives many advantages in terms of consumption of fuel and tires, safety, ergonomic functions and drive comfort. In particular, the SBW technology improves the safety functions thanks to fault-tolerant electronic distributed system; the absence of the steering column allows to realize safer cabines, with alternative structures that withstand more strict crash-test and also enables easy changes to pass from "left-drive" to "right-drive". Moreover, thanks to the absence of a direct mechanical link between the steering wheel and the wheels, the vibrations (i.e. due to the road surface or to strong cross-wind) are eliminated.

It is also known that the benefit of a “steer-by-wire” technology is greater if a vehicle adopts many other “x-by-wire” technologies: brake, throttle, active suspension and so on.

This long term perspective leads to the so called “corner-by-wire” vehicle. The use of “by-wire” technology is not limited to road vehicles. In particular, it can give even more advantages in industrial and off-highway vehicles, too.

The final goal of this project is to realize an Electro-Hydraulic SBW system (EH-SBW) for off-highway vehicles with commercial components. In particular, with a compact, modular and cheap electro-hydraulic steering rod actuator, active force-feedback and low cost microcontrollers as core of the electronic control units (ECUs) of the system.

Homologation laws for off-highway vehicles are less strict than for road cars (speed constraints, for example), therefore some purely hydraulic steering systems (i.e. steering column substituted by hydraulic pipes) without force-feedback are already available on the market.

With negligible modifications on the steering rod of a pure hydraulic steering system, a compact electro-hydraulic rod actuator can be realized. A complete SBW system can be made by adding a force-feedback actuator on the steering wheel.

For this reasons, it is reasonable to think as EH-SBW systems promise to be implemented in the near future, before SBW for road vehicles.

In this paper we focus on the force-feedback system (FFS) of a generic SBW system, in particular on the fault tolerance and the self-diagnostic processes of the system.

Since we describe a force feedback system for a generic SBW application, the following considerations are valid both for off-highway (and industrial) vehicles and for road vehicles.

Fault Tolerance Techniques

Two principles must be considered in a fault tolerant design. The first one is the so called “error processing” and the second one is the “fault containment”.

Error processing aims at removing errors from the computational state, if possible before a failure occurs; there are several techniques to achieve this goal, like error detection (to detect errors), error masking (to mask the effects of errors), error recovery (to restore the system to an error free state).

Fault Containment aims at preventing faults from affecting other (redundant) units in the system. A fault tolerant system consist of several fault containment regions.

Fault tolerance methods generally use redundancy. This means that in addition to a specific module, one or more modules are connected, usually in parallel.

Redundancy can be implemented in hardware, in information, in time and in software.

Concerning hardware redundancy the basic idea is to overcome hardware faults by using additional hardware. This is the simplest redundancy technique and due to the decreasing costs of the components it is a feasible solution, too.

Three different approach to hardware redundancy can be identified: passive, active and hybrid.

A passive redundancy allows to achieve fault tolerance without any action. In its simplest version there is no error detection and the fault is masked exploiting the voting algorithm between the results produced by redundant modules. Active redundancy is a dynamic technique and requires fewer modules at the cost of an increased information processing cost. Usually, it detects existence of fault and performs a reconfiguration of the system to remove faulty modules.

The hybrid approach is the combination of passive and active redundancy.

As described in the next section, our system propose a hardware redundancy with some elements of software redundancy.

The Redundancy Architecture Used

Fig.1 shows a simplified block diagram of the proposed architecture. In this solution we propose a Triple Modular Redundancy (TMR) architecture to determine the position of the motor shaft of the BLDC motor used as force feedback actuator (see next section).

Using this technique a single faulty module is tolerated without any additional information on the specific faulty module.

The fault can be functional or technological, single or multiple, but must remain confined in this faulty module.

With TMR if one module becomes faulty the other two mask the fault by means of a voting algorithm.

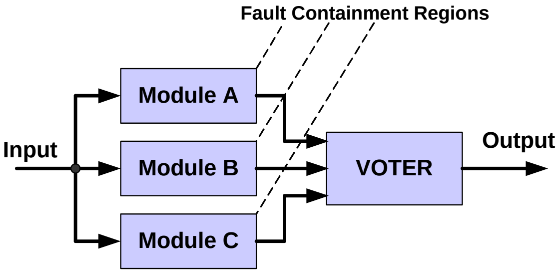

Fig.1 TMR Architecture

From an hardware point of view TMR can be applied at different levels. For example, it can be applied at application level (tripling the whole system) or at system level (tripling the processor), like in our system.

We have used three different processors, as it will be described in next section. Consequently, due to the different capabilities of the processors, the determination of the position is performed by three different software algorithms.

Clearly, in a TMR architecture if the voter fails a complete system fault occurs, therefore we have implemented a simple self diagnostic process of the system.

In particular, the voter recognizes if a position data coming from one of the three redundant systems is corrupted and generates the corresponding error code.

Summarizing, we have adopted a hybrid approach: the proposed system is a system with complete passive redundancy and some elements of active redundancy. In fact, the system perform an error detection but not a reconfiguration.

Hardware-in-the-Loop Approach

In this paper a Hardware-in-the-Loop approach (HIL) has been used.

HIL refers to a technology where some of the components of a model (virtual hardware) are replaced with real hardware at different steps. The goal of this procedure is the replacement of all virtual hardware components with real ones to obtain a complete real system with the same behaviour of the initial virtual system.

We use HIL to test the prototypes of the developed electronic control unit and its control algorithms, and to test the real time behaviour of hardware components.

This technology provides a way for testing control systems over the full range of operating conditions, including failure modes. Testing a control system prior to its use in a real application can reduce the cost and the development cycle of the overall system.

HIL gives others benefits like the realization of fewer prototypes with less test drives (with consequently lower costs), or a more systematic tests in a shorter time with higher quality and lower risks, too.

Description of the Force Feedback System

A simplified block diagram of the implemented force feedback system is shown in Fig.2.

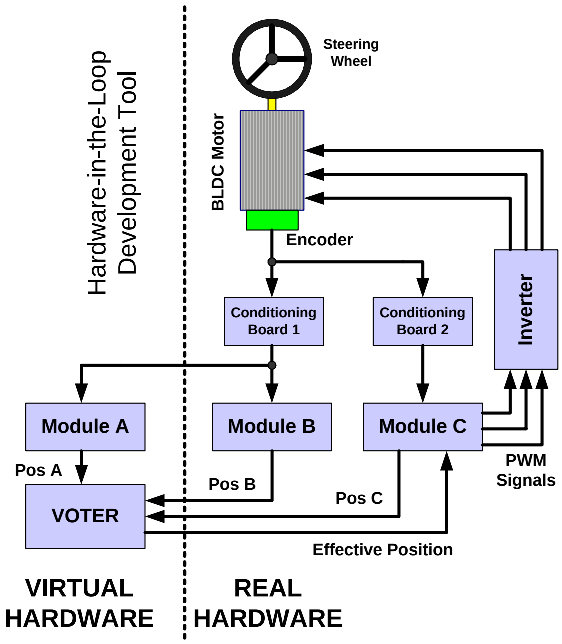

Fig.2 Simplified Block Diagram of the Implemented Force Feedback System

A steering wheel has been directly connected to the motor shaft of the BLDC motor used as force feedback actuator.

An encoder is integrated on the motor shaft, too.

The output position signals from the encoder are input to two conditioning boards to adapt the signal levels to the format requested by the modules A, B and C. The two boards use the same basic principles but are implemented with different hardware.

A is the first module of the TMR architecture and it is implemented by the virtual hardware development tool. **B** is the second module of the TMR architecture. It is a real module that estimates the position of the motor shaft exploiting a 8-bit microcontroller core. **C** is the third module of the TMR architecture. It is a real module that estimates the position of the motor shaft exploiting a 16-bit microcontroller core.

The three estimated positions are used by the voter, that is implemented by the same HIL development tool used to implement module A.

In addition to position calculation module C implements also the torque control algorithm exploiting the position estimated by the virtual voter.

The BLDC motor needs an inverter to operate correctly. The inverter inputs are the three PWM signals coming out from C; inverter output is the three-phase power supply needed to control the BLDC motor.

The inverter is designed to operate both with a 12Volt power supply, like the one used in today vehicles, and 42Volt power supply, the future standard.

All the real hardware for the system has been designated and implemented in prototypes assembled in the labs of our University.

Voting Algorithm

As described in the previous section, the output position signals from the encoder are input to two conditioning boards to adapt the signal levels to the format requested by the modules A, B and C. The two boards use the same basic principles but are implemented with different hardware.

Due to the different computation strategies, the three position values estimated by the modules can be different (within a well defined range) also in absence of failures.

For this reason, the classic “Majority Voting” algorithm is not suitable for a TMR architecture like the one here proposed.

Therefore a more complex voting strategy has been implemented. In particular, the voter implements a “Mid-Value Select” technique (MVS).

At each instant, the algorithm sorts the position signals estimated by the three modules A, B, C and selects the median value. This value is the real position used in the control algorithm of the BLDC motor. As mentioned above the control algorithm is performed by C.

With a high probability, in case of a failure in one of the three modules, the corresponding data is corrupted and very different from the others. Using a MVS strategy the corrupted data is automatically discarded because it fills the lower place or the upper place in the list of sorted position data.

If a node is detected as faulty, the voting algorithm computes the real position as the mean value of the two remaining nodes and uses the half adjust rules.

As already previously, the voter implements a simple self diagnostic process, too.

For each module of the TMR architecture the voter can recognize if the corresponding position signal is corrupted.

For each position signal coming from a module (A, B or C) the voter checks if this signal is included in a dynamic range of the last real position estimated by the voting algorithm.

If the comparison gives a negative result for a defined number of consecutive times, then a failure is occurred in the corresponding module.

To notify this self diagnostic information, the voter sends a status system message on the bus.

An univocal warning code is assigned to each module. Moreover two other codes are generated by the voter: the first one identifies the correct operational condition of the system, the second one is generated when more than one module is detected as faulty and is a real error code.

These five codes (fault on A, fault on B, fault on C, system ok and system error) are generated by the voter by means of a simple combinatorial logic function.

The CAN Network

As shown in Fig.3 the CAN network is composed by four nodes. The traffic on the bus is monitored by the monitor node implemented in hardware by a dedicated development tool. The position signals estimated by B and C are sent to the virtual voter implemented in A. The voter sends the real position to C that uses this information to perform the torque control algorithm and drive the motor by means of the inverter board.

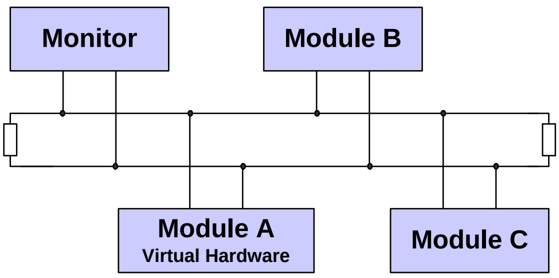

Fig.3 CAN Network

As mentioned above, the voter performs the self diagnostic process, too. Therefore the voter sends on the bus a message containing the state of the complete system.

In this prototypal architecture this message is received by the monitor node, but in a complete system (on a vehicle) this message can be received by other nodes (ECUs) that can take counter-measures to lead the system in a safe state in case of failure of more than one module of the TMR architecture.

Evidently, the real position can be used by other nodes to improve performance and safety of the vehicle, too.

In safety critical applications, like steer-by-wire, event-triggered protocols like CAN are not suitable due to their non deterministic behaviour.

In this preliminary architecture that exploits an HIL approach, a dedicated network has been implemented and therefore the disadvantages of the event-triggered protocol are considerably reduced. Moreover the main goal of this work was to validate the algorithms implemented by A, B, C and verify the voting algorithm using a well know basic torque control algorithm performed by C.

Experimental Results

The Hardware-in-the-Loop system described previously has been used also to test and validate the procedure to determine the position of the motor shaft by B and C. The voting algorithm has been tested and validated.

To perform the tests a basic torque control strategy based on the difference between the position of the motor shaft and the wheels position has been used (see [6]).

The same tests are executed at different baudrates. At low baudrates (100kBaud) the bus load is higher than in the case of high baudrates (i.e. 1MBaud) but no meaningful modifications in the system behaviour can be detected.

In the following, the experimental results with a baudrate of 1MBaud are presented.

To validate the entire system we need two additional data: the wheel position and the steering stiffness. This data are provided by the virtual hardware (HIL development tool).

The steering stiffness is an electronic parameter that reproduces the stiffness of the steering column present in the traditional steering system in a steer-by-wire system.

In Fig. 4 a typical result of a HIL experiments is shown. A sinusoidal wave is provided as wheels positions (dashed line) and one can see that the position estimated by the TMR architecture (solid line) follows exactly the wheels reference signal. This means that the control algorithm works correctly, too.

Due to data processing time a minimum delay can be noted.

Fig. 5 shows the estimated position and the single components that are used to calculate the mid-value. It is possible to note that the estimated position calculated by the system (solid line) derives from the mid value select strategy implemented by the voter. The dashed line is the position estimated by module A, the dash-dotted line is the position estimated by module B and the dotted line is the position estimated by module C.

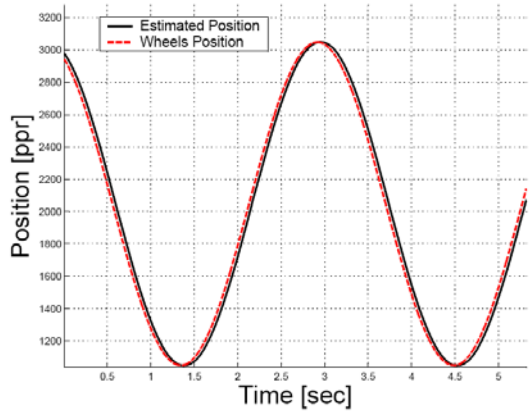

Fig. 4 Tracking Test: Wheels Position (Reference, dashed line) and Estimated Position (calculated, solid line)

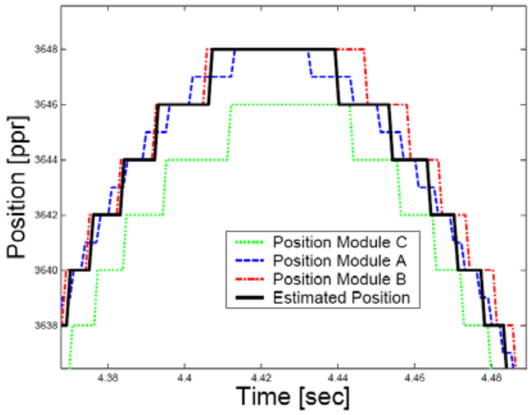

Fig. 5 Detail of the Signals used to calculate the Estimated Position.

The encoder of the BLDC motor used is composed by an incremental encoder and an absolute encoder. Due to the limitation of the HIL development tool only the signal coming from the incremental encoder has been used and the unit of measurement are pulse per round (ppr). 4096 pulses correspond to 360 mechanical degrees of the motor shaft.

The three modules produce very similar results. In many conditions the differences are below the resolution, therefore half adjust phenomenon can occur and the real position seems to mismatch with the rules of the voting algorithm.

If we change some system parameters, like the steering stiffness, we change also the system behaviour. This is shown in Fig. 6.

With low stiffness the system is not very reactive and it introduces some delay to follow the reference signal. On the contrary, with high steering stiffness the system is more reactive, but it introduces some overshoot effects due to the mechanical inertia of the steering wheel.

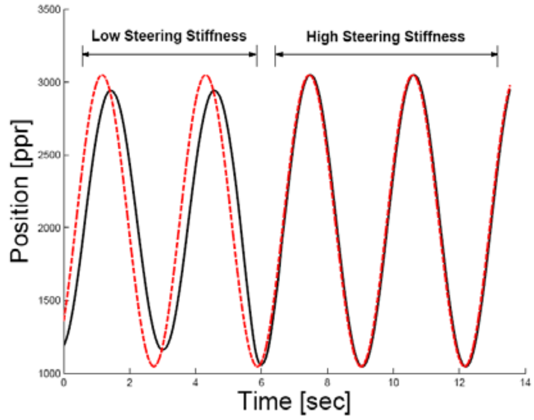

Fig. 6 Estimated Position with Different Steering Stiffness Values

The effects of the stiffness variation is more evident if a pulse sequence is provided as reference wheels position signal, shown in Fig.7.

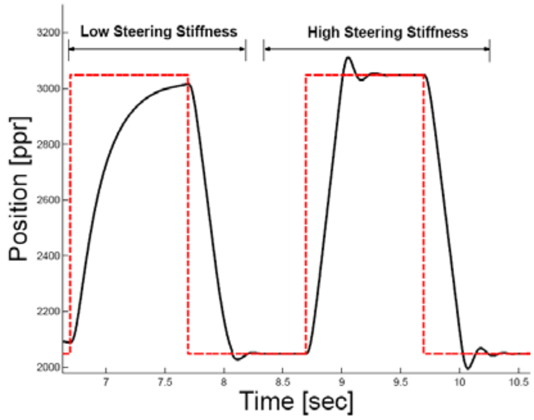

Fig.7 Estimated Position with different Steering Stiffness Values, when a Pulse Wave is used as Input

This effect can be considerably reduced using a more complicated torque control algorithm that uses a current closed loop technique on the currents of the three phases of the BLDC motor (see [6]).

This algorithm requires an additional hardware to acquire the currents and it is not considered here because the main goal of this work was to validate the proposed TMR architecture and to verify the correctness of the voting strategy and of the position detection algorithms of modules B and C.

All these considerations are valid if all the modules and the voter of the TMR architecture are working correctly, as notified by the voter itself with the system status message (see Fig.8).

Fig.8 System Status Message Display-System OK

A fault injection has been performed to complete the validation of the system. In Fig. 9 a fault in module B has been introduced.

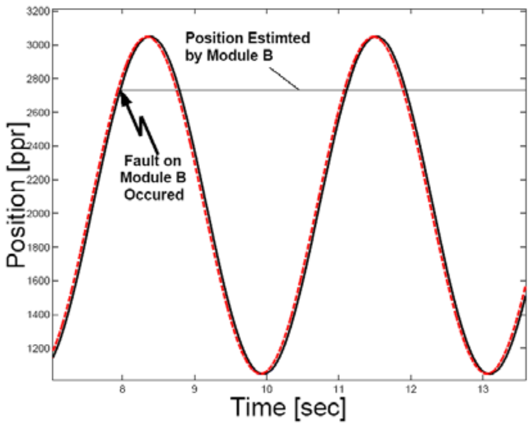

Fig. 9 Fault Injection on Module B

The system still works correctly, but changes the voting technique: the mid value select technique is substituted by a classic mean value technique, and the voter modifies the contents of the system status message as described previously and shown in Fig.10.

Fig.10 System Status Message - Module B does not work correctly

To show the impact of this change in the voting strategy, Fig. 11 shows the signals after the time when the fault is injected in Fig. 9.

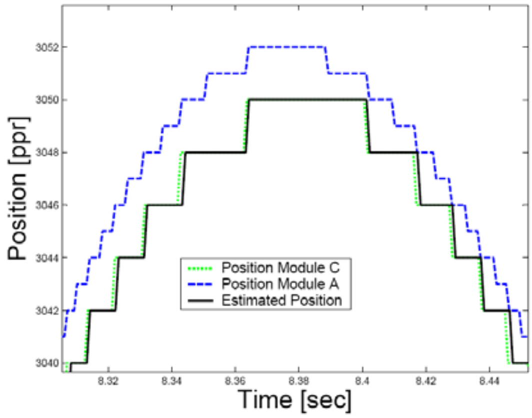

Fig. 11 Detail of the Signals used to calculate the Estimated Position after the Fault on Module B.

Due to the resolution limits, the same half adjust phenomenon described previously occurs again.

Of course, if more than one module is faulty, they are detected by the voter. The system still works using as input the only operating module, but the voter includes this system error code in the system status message sent on the bus.

Fig.12 System Status Message – Module B and Module C do not work correctly

Fig.12 shows an example of the system status message when a double fault on module B and module C is detected.

Conclusions

In safety critical applications like steer-by-wire, a deterministic behaviour of the system is needed, and an implementation based on time triggered protocols is required. Therefore from a logic point of view an implementations based on an event triggered protocol like CAN is not recommended. But the main goal of this work was to verify the strategies of determination of the position of the motor shaft of the BLDC motor using an 8-bit architecture and a 16-bit architecture.

In this case, the disadvantages of the event triggered protocols are considerably reduced due the implementation of a dedicated network.

The results of HIL experiments show how the implemented algorithms are correct and the behaviour of the system is still valid in case of a single faulty node, too.

This means that the proposed TMR architecture can be used as starting point to implement also a complete software redundancy and a redundant architecture more complex than the TMR solution we adopted. In particular, since the correctness of the voting algorithm has been verified, future developments include the implementation of a redundancy of the voting subsystem.

References

[1] Fault-Tolerant Drive-by-Wire Systems *Isermann, R.; Schwarz, R.; Stolzl, S.;* Control Systems Magazine, IEEE , Volume: 22 , Issue: 5 , Oct. 2002, Pages:64 – 81.

[2] Real-Time Systems, Design Principles for Distributed Embedded Applications *Kopetz H.;* Kluver Academic Publisher, ISBN 0-7923-9894-7.

[3] OSEKtime: A Dependable Real-Time Fault Tolerant Operating System and Communication Layer as an Enabling Technology for By-Wire Applications *Poledna S., Glück M., Tanzer C., Boutin S., Dilger E., Führer T., Ebner C., Fuchs E., Belschner R., Hedenetz B. Holzmann B., Schedl A., Nossal R., Pfaffeneder B., Ringler T., Domaratsky Y., Krüger A., Zahir A.;* in SAE paper 2000-01-1051, 2000.

[4] A System Safety Process for By-Wire Automotive Systems *Amberkar S., D'Ambrosio J.G., Murray B.T., Wysocki J., Czerny B.J.;* in SAE paper 2000-01-1056, 2000.

[5] Reliability Prediction of Fault Tolerant Automotive Systems *Theis I., Guldner J., Orend R.;* in SAE paper 2000-01-1049, 2000.

[6] Design and Implementation of a Steer-by-Wire System for Off-Highway Vehicles *Bertacchini A.*,Ph.D. Thesis 2004.

Alessandro Bertacchini
University of Modena and Reggio Emilia
DISMI
Via Fogliani,1 – 42100 Reggio Emilia, Italy
Phone: +39 059 2056169
+39 0522 522625
Fax: +39 059 2056129
E-mail: bertacchini.alessandro@unimore.it
Website: www.elettronica.ingre.unimore.it

Luca Tamagnini
University of Modena and Reggio Emilia
D.I.I.
Via Vignolese, 905/b
41100 Modena,Italy

Matteo Mistrorigo
University of Modena and Reggio Emilia
DISMI
Via Fogliani,1 – 42100 Reggio Emilia, Italy

Prof. Paolo Pavan
University of Modena and Reggio Emilia
DISMI
Via Fogliani,1 – 42100 Reggio Emilia, Italy
Phone: +39 059 2056158
+39 0522 522605
Fax: +39 059 2056129
E-mail: paolo.pavan@unimore.it
Website: www.elettronica.ingre.unimore.it