ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1988
Seria: AUTOMATYKA z.96 Nr kol.972

Aleksander Staszulonek
Politechnika Śląska

JOINT MOTION SYNCHRONIZATION IN TRACKING ROBOTS

Abstract:

The presented paper gives a report on research concerning the synchronization of manipulator motion in advanced controllers able to track the moving objects. The idea of control hardware is presented together with the analysis of expected static and dynamic tracking errors. The importance of velocity feedforward has been stressed and in depth discussed. Chosen block diagrams of the control software implementing the discussed ideas are also presented.

1. Introduction

One of the essential problems accompanying the design of controllers for the tracking robots is the synchronization of motion of the robot's joints. Another of the problems is the necessity to eliminate the transient processes during the acceleration of joints. To achieve this a special algorithms generating the smooth acceleration profile have to be implemented. As the example a two joint structure has been taken and the control hardware and software enabling the accurate tracking control of the robot arm have been designed and tested

The controller that has been developed fulfils the following requirements:
- setting of the correct transmition ratio,
- setting of the correct rotation speed, based on the given,
- control of the accelerations during speed changes and keep perfect synchronization during accelerations,
- display of the most important system parameters on a terminal via the serial port.

The control algorithm, and created software are presented in the subsequent chapters of this work.

2. Principles of controller operation

From the theory point of view the control problem is the following: the position of two independent axes has to be controlled simultaneously and synchronized according to a

reference position value generated by the velocity integrator. The integration rate is given by a constant velocity factor during normal operation. Two inputs determine the desired velocities of both axis. The ratio between the velocities of each axis is determined by the higher level software receiving the information about the position and orientation of the object to be tracked. The controller should guarantee accurate tracking of the objects moving with constant acceleration.
To guarantee smooth transition from one velocity to another an acceleration profile has to be generated during the transient for both axes. This acceleration control provides good behavior during speed change and will prevent excitation of the mechanical structure. Maximum error between both axis should be less then
0.1 mm.

The general lay out of the controller is given on fig. 2.1

The controller can be divided into analog and digital parts. The digital part consists of a software PID position controller, an optical incremental encoder ENC as the position sensor, software velocity integrator, position differentiator, acceleration profile controller. The analog part consists of a closed loop VELOCITY CONTROLLER using tachogenerator feedback.

2.1. Description of the digital control algorithm.

The reference position for both axes is obtained by the integration of the desired velocities.

The present position is obtained from the encoder interface. Then for each sampling instant i the position error is calculated.

$$e_i = x_{ri} - x_i \qquad (2.1)$$

where

e_i is position error
x_{ri} is reference position
x_i is present position
i is the sample number.

The present velocity v_{mi} is calculated as the difference between actual and previous positions for a fixed time difference and is expressed in number of pulses per sample.

$$v_{mi} = x_i - x_{i-1} \qquad (2.2)$$

The reference velocity v_{rfi} is also expressed in number of pulses per sample and can be defined as

$$v_{rfi} = x_{ri} - x_{ri-1} \qquad (2.3)$$

The differential part of the controller is obtained from the following formula:

$$de_i = e_i - e_{i-1} \qquad (2.4)$$

which can also be expressed by

$$de_i = (x_{ri} - x_{ri-1}) - (x_i - x_{i-1}) \qquad (2.5)$$

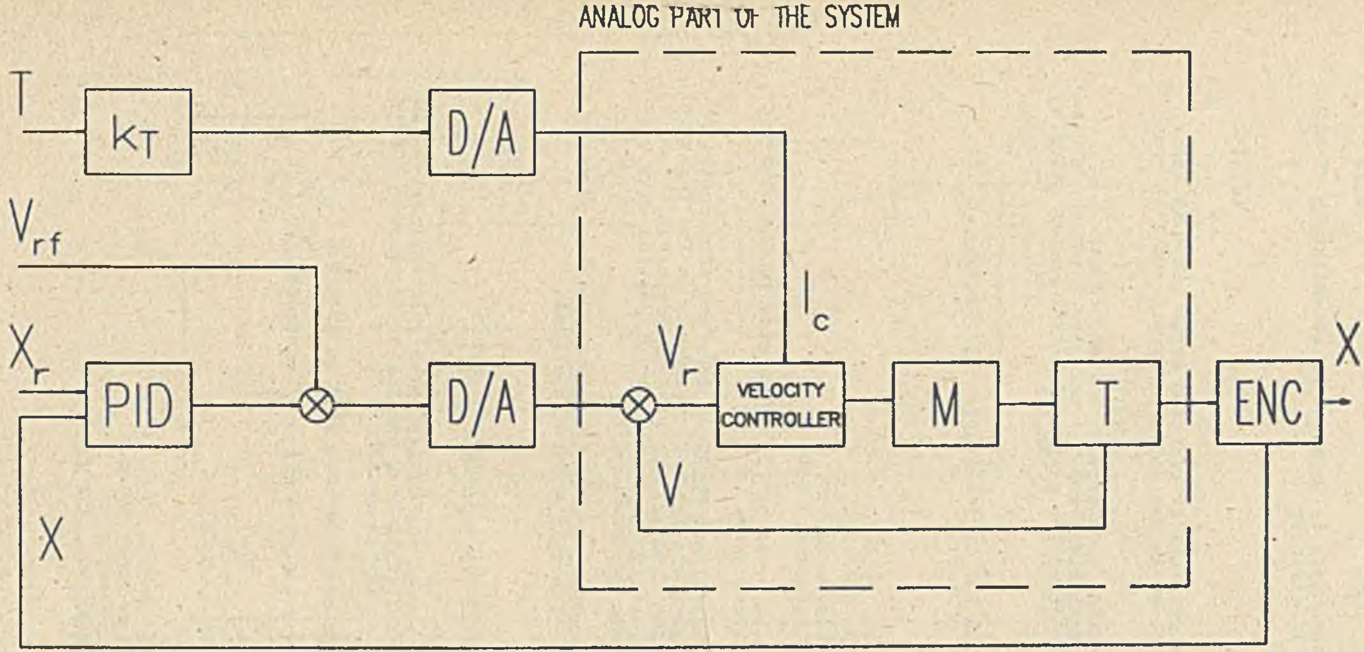

Fig. 2.1. GENERAL CONCEPT OF CONTROLLER

Rys. 2.1. Ogólna idea regulatora

Thus the differential part can be presented in the form

$$de_i = v_{rfi} - v_{mi} \tag{2.6}$$

The integration part is obtained by accumulation of all past errors and is given by

$$I_i = T \sum_{k=0}^{i} e_k \tag{2.7}$$

where I_i, is the integral of position error and T is the sampling interval, e_k is the position error, and k is the summation index

These expressions constitute the standard PID controller. After introducing the velocity feedforward term the control law applied for position control can be given by equation

$$PIDF = k_p e_i + k_s T \left(\sum_{k=0}^{i} e_k \right) + k_d (v_{rfi} - v_{mi}) + k_{ff} v_{rfi} \tag{2.8}$$

where:

k_p is proportional gain,
k_s is integration factor,
k_d is differentiation factor,
k_{ff} is velocity feedforward gain.

The control law given by (2.8) has been implemented for both axes in software form.

The desired velocities obtained from the PIDF position controllers are sent via the D/A converters to the inputs of the analog velocity controllers with tachogenerator feedback. This part of the system is implemented with the commercial Infranor servocontrollers.

Problems concerning the disturbing torque compensation are dealt with in the following subsections.

2.2. Analysis of static and dynamic errors.

To evaluate static and dynamic errors in the system, an approximate block diagram of a controller for a single axis can be given as presented on fig. 2.2.

The following symbols have been used on fig. 2.2.

$K_o(s)$ is the motor velocity transfer function and

$$K_o(s) = \frac{k_o}{1+sT_o} = \frac{V(s)}{U_m(s)} \tag{2.9}$$

where
$V(s)$ is the motor speed,
$U_m(s)$ is the motor input voltage,
k_o is the motor gain,
T_o is the time constant of motor.

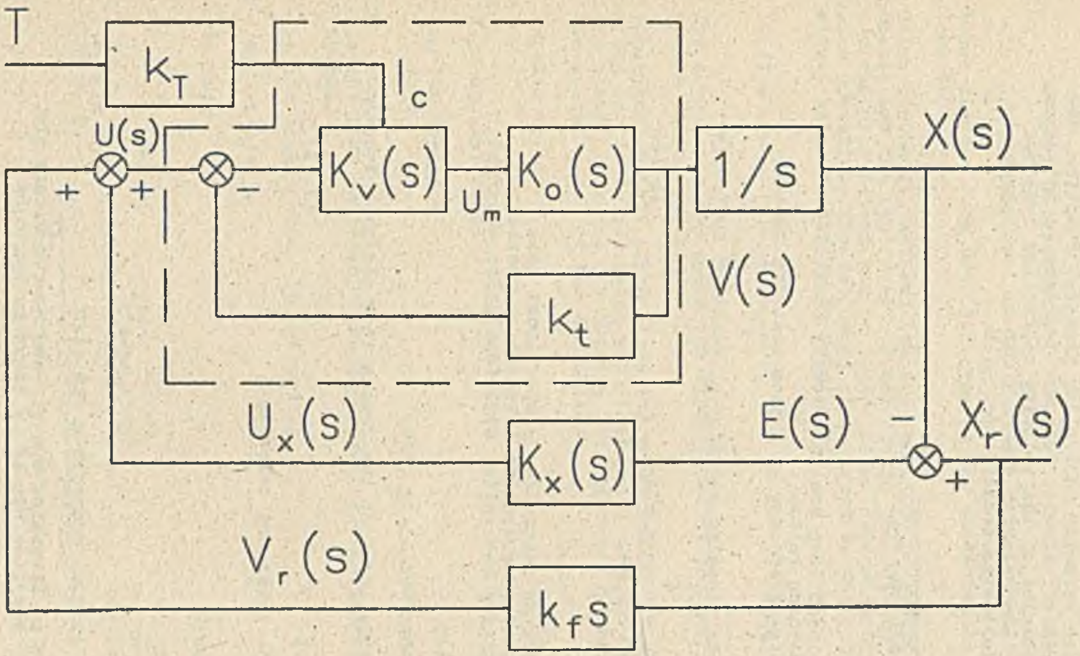

Fig. 2.2. BLOCK DIAGRAM OF CONTROLLER.

Rys. 2.2. Schemat blokowy regulatora

The velocity to position transfer function is expressed by

$$K_p(s) = \frac{1}{s} = \frac{X(s)}{V_m(s)} \tag{2.10}$$

where

X(s) is the motor position.

$K_v(s)$ is the velocity controller transfer function. The exact expression of velocity transfer function is not given, however the behavior of the velocity controller together with the motor and velocity feedback can be approximately given by the following transfer function

$$K(s) = \frac{k}{s^2+2\varepsilon\omega_n s+\omega_n^2} \tag{2.11}$$

where

k - is the closed loop analog velocity controller gain,
ε - is the damping ratio,
ω_n - is the eigenfrequency of analog velocity control loop.

$K_x(s)$ is the transfer function of position controller

$$K_x(s) = k_x + \frac{1}{sT_{ix}} + sT_{dx} \tag{2.12}$$

where

k_x - is position controller proportional gain,
T_{ix} - is position controller integration constant,
T_{dx} - is position controller differentiation constant

and additionally

k_f - is feedforward loop gain.

The reference input of the system is a position function given by

$$x(t) = v_r t \tag{2.13}$$

where v_r is the reference velocity.

In Laplace domain

$$X_r(s) = \frac{v_r}{s^2} \tag{2.14}$$

assuming that initial conditions are zero.

To be able to determine dynamic and static errors the error transfer function has to be found in the form

$$E(s) = K_e(s)*X_r(s) \tag{2.14a.}$$

For this reason the following set of equations can be written based upon the block diagram from fig. 2.2.

$$X(s) = \frac{1}{s} V(s) \quad (2.15)$$

$$V(s) = K(s)*U(s) \quad (2.16)$$

$$U(s) = U_x(s) + V_r(s \quad (2.17)$$

$$V_r(s) = k_f*s*X_r(s) \quad (2.18)$$

$$U_x(s) = K_x(s).E(s) \quad (2.19)$$

$$E(s) = X_r(s) - X(s) \quad (2.20)$$

Block diagram from fig. 2.2. can be now presented as given on fig. 2.3.

Using previously defined transer functions of system's components and the equations (2.15)-(2.20) the desired error transfer function can be found. After some elementary symbolic operations the following error transfer function of the single axis controller is obtained:

$$E(s) = \frac{1 - K(s)k_f}{s + K(s)K_x(s)} * \frac{1}{s} v_r \quad (2.21)$$

The above equation gives the dynamic errors present in the system during the transient.

Substituting full expressions for the transfer functions appearing in (2.21) and using the final value theorem for Laplace transformations one obtains the following expression for the static error of the controller

$$e_s = \lim_{t - \infty} e(t) \quad (2.22)$$

which is equivalent to

$$e_s = \lim_{s - 0} s*E(s) \quad (2.23)$$

where
$e(t)$ is the time representation of dynamic errors in the system,

e_s is the static error in the system,

and $E(s) = L\{e(t)\}$.

In full form

$$e_s = \lim_{s - 0} v_r \frac{1 - \dfrac{kk_f}{s^2+2\varepsilon\omega_n s+\omega_n^2}}{s + \dfrac{k}{s^2+2\varepsilon\omega_n s+\omega_n^2}\left(k_x + \dfrac{1}{sT_{ix}} + sT_{dx}\right)} \quad (2.24)$$

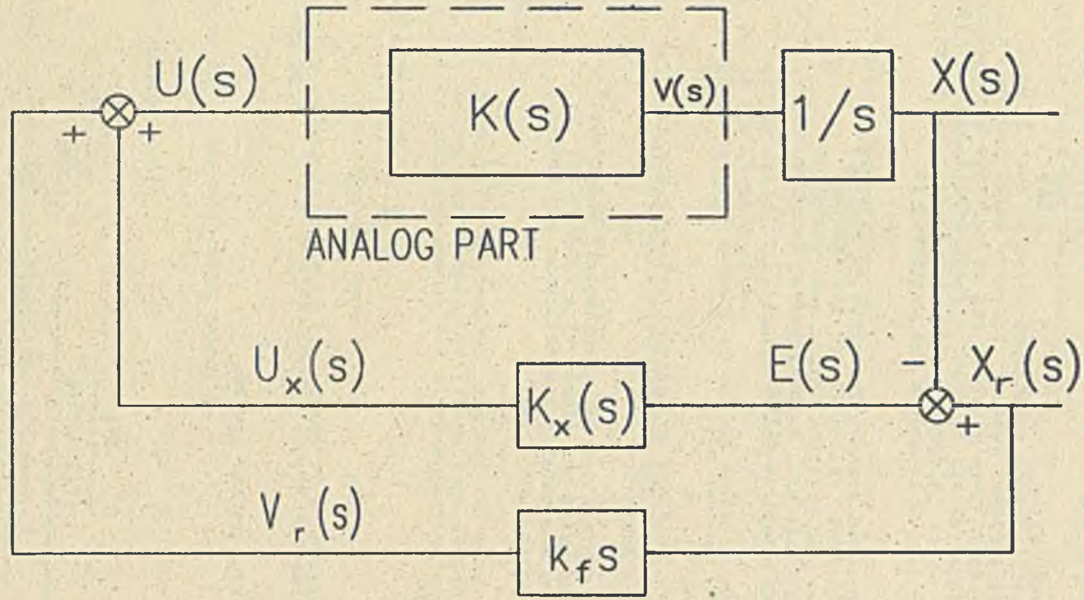

Fig. 2.3. REDUCED BLOCK DIAGRAM OF THE CONTROLLER

Rys. 2.3 Zredukowany schemat blokowy regulatora

To interprete this limit correctly the implementation aspect of controller has to be considered. If there would be no limits on the integration part of control signal, then for the position control with constant velocity input the equation (2.24) takes form

$$e_s = \lim_{s \to 0} v_r \frac{sT_{ix}\left(1 - \dfrac{kk_f}{s^2+2\varepsilon\omega_n s+\omega_n^2}\right)}{s^2T_{ix} + \dfrac{k(1+sk_xT_{ix}+s^2T_{ix}T_{dx})}{s^2+2\varepsilon\omega_n s+\omega_n^2}} = 0 \quad (2.25)$$

For the positon control with constant acceleration input the equation (2.25) takes form

$$e_s = \lim_{s \to 0} v_r \frac{T_{ix}\left(1 - \dfrac{kk_f}{s^2+2\varepsilon\omega_n s+\omega_n^2}\right)}{s^2T_{ix} + \dfrac{k(1+sk_xT_{ix}+s^2T_{ix}T_{dx})}{s^2+2\varepsilon\omega_n s+\omega_n^2}} \quad (2.26)$$

resulting finally in equation

$$e_s = \frac{(\omega_n^2 - kk_f)T_{ix}v_r}{k} \quad (2.27)$$

It can be seen from (2.27) that there will be no static error in the system during the acceleration if only the feedforward gain is chosen so that

$$kk_f - \omega_n^2 = 0 \quad (2.28)$$

The above formulas are true only if there are no limits on the value of integration part of the control signal. Practically the integration part of controller is always limited.

Therefore a different case has to be considered when the integral reaches the limit value. If the integrator becomes saturated the structure of the controller is changed - there is practically no integration present in the system. To reflect this situation a maximal absolute value of the integration signal I_{max} is substituted into the equation (2.24) so that the following equation is obtained

$$e_s = \lim_{s \to 0} v_r \frac{1 - \dfrac{kk_f}{s^2+2\varepsilon\omega_n s+\omega_n^2}}{s + \dfrac{k}{s^2+2\varepsilon\omega_n s+\omega_n^2}(k_x+I_{max}+sT_{dx})} \quad (2.29)$$

which gives as the result

$$e_s = v_r \frac{\omega_n^2 - kk_f}{k(k_x+I_{max})} \quad (2.30)$$

From the last equation one can see that if the value of integration in the controller is limited the condition

$$kk_f-\omega_n^2=0 \tag{2.31}$$

must be already satisfied for the position control with constant velocity input. This shows that the choice of the feedforward gain factor k_f is crucial for the quality of control. Since the value of k_f is digital, the quantisation error is present in the system. To eliminate this disadvantage an additional value representing the integration part average value over the period of last 64 samples, is added to the value of control signal. This prevents the integrator from entering the saturation as long as the value of feedforward gain does not deviate significantly from the correct one.

2.3. Acceleration profile control.

To achieve a smooth acceleration profile during transition from one velocity to another, a second derivative of acceleration is generated. The acceleration absolute maximum value is limited to a specified value, desired velocity is achieved with zero acceleration at the end point of transient. The profile of the second derivative of acceleration, necessary to satisfy these requirements is presented on fig. 2.4a.

After first integration, the first derivative of acceleration is obtained as shown at fig. 2.4b.

The second integration gives the required smooth acceleration profile curve (fig. 2.4c.) consisting of parts of parabolas and flat line when the acceleration is maximal.

After the third integration a reference velocity profile during speed change is obtained as shown on fig. 2.4d.

Assuming the pulse duration t_1 of acceleration second derivative, and maximal acceleration value a_{max}, it is possible to calculate the required values of constant acceleration period t_2 and the amplitude of acceleration second derivative necessary to achieve the desired end velocity.

Simple integration shows that by using this method the end velocity will be equal to

$$v_1 = \alpha t_1^2(t_2+2t_1)+v_o \tag{2.32}$$

where α is the second derivative of acceleration.

The maximal acceleration during transition will be equal

$$a_{max} = \alpha t_1^2 \tag{2.33}$$

Having already assumed t_1 and a_{max} one obtains

$$\alpha = \frac{a_{max}}{t_1^2} \tag{2.34}$$

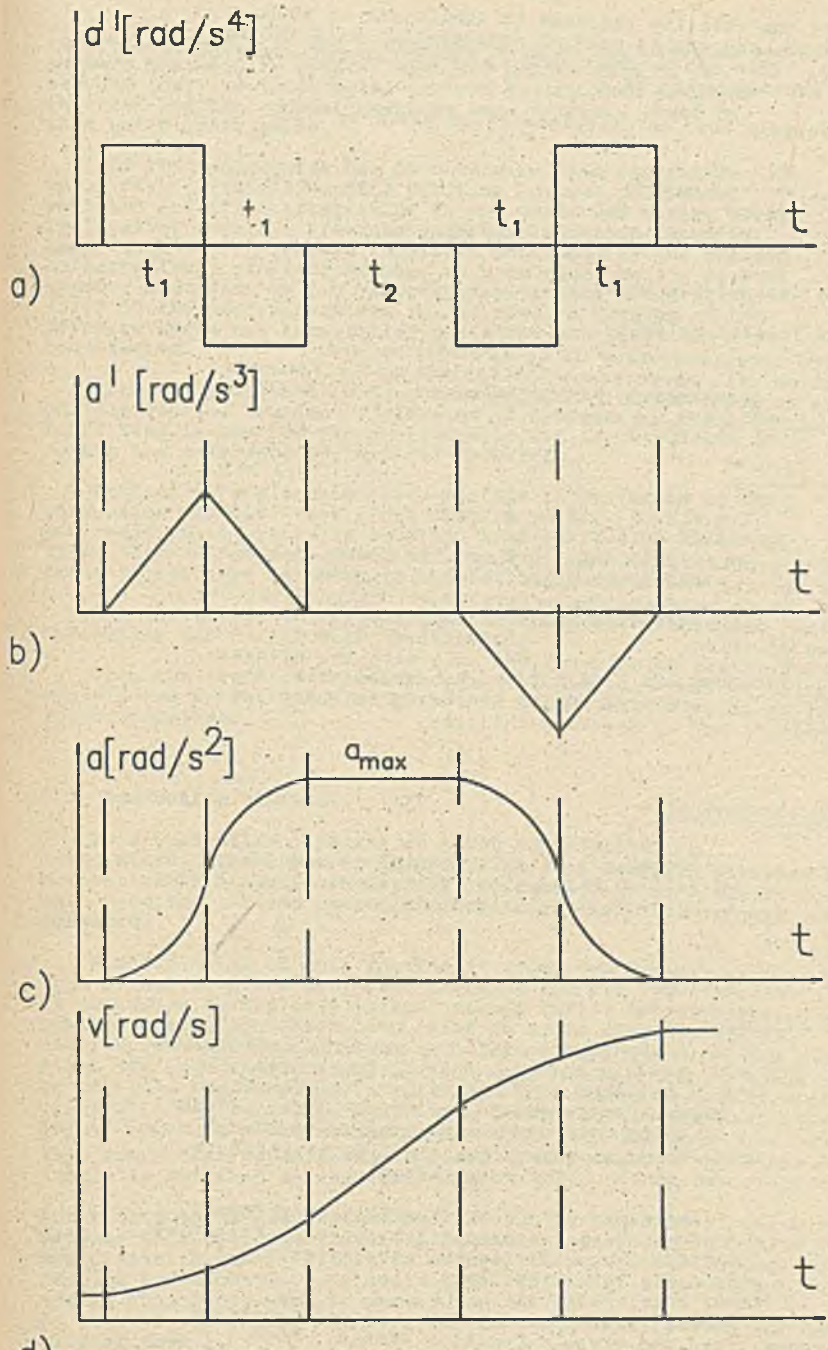

Fig. 2.4. ACCELERATION PROFILE GENERATION

Rys. 2.4. Generowanie profilu przyspieszenia

However if the velocity increase is small, and t_1 remains always fixed, the use of maximal acceleration will not be necessary. This can be seen when t_2 is calculated using the obtained maximal value of α .

The formula used to calculate t_2 is given by

$$t_2 = \frac{v_1 - v_0}{\alpha t_1^2} - 2 t_1 \qquad (2.35)$$

where

v_1 is the desired end velocity,

v_0 is the initial velocity.

If the resulting value of t_2 is negative the acceleration will never reach the maximum and it becomes necessary to recalculate α using equation (2.34) with $t_2 = 0$ which results in

$$\alpha = \frac{v_1 - v_0}{2 t_1^3} \qquad (2.36)$$

Such a procedure has been implemented by the control software for the chosen example. It requires three integrations to generate the reference velocity during the transition. One more integration is necessary to obtain the reference position.
To calculate these parameters, the axis with the highest acceleration is choosen. Then, for the second axis, the parameter α_2 is calculated based on the α_1 of the first axis, while t_1 and t_2 are kept the same.

3. Basic subroutines.

The software designed for the presented controller is interrupt driven with the programmable timer interrupt being the event synchronizing the work of the system.

3.1. Subroutine CONTROLLER

Routine operation

After the interrupt service routine has been called, the present state of the registers is saved on the stack, the data block PAR1 base address is loaded into A4, the data block PAR2 base address is loaded into A5, I/O base address is loaded into register A1, timer base address is loaded into A0 and timer interrupt request flip-flop is reset so that counters can continue their operation.

The delay introduced here by the operations performed between interrupt occurrence and interrupt request flip-flop reset is compensated by specifying the shorter sampling interval during timer initialization.

Then the present values of encoder interface counters are latched, read and present position stored. Both counters are latched simultaneously.

When the present position is stored, it has to be compared with previous encoder position to check whether the hardware reset has occurred during the last sampling interval or not. If the difference between the previous and present position is greater then 200 pulses then it is assumed that the index pulse arrived during last sampling interval and the encoder counters were hardware reset by this gated index pulse.

If the index pulse has been received the controller has to be reset. Present encoder position becomes reference position (position integration is not performed during first cycle after reset). Previous reference position value is made equal to the present reference decreased by the desired velocity (VD), previous encoder is made equal to the present encoder decreased by the velocity desired and integration parts of the controllers are set to zero. A version of the software where the integration parts are not reset has also been tested.

Then the acceleration flag is checked and acceleration profile control routine is executed if necessary. Next the RESET flag is checked and desired velocity is integrated to obtain new reference position if necessary.

Control of the acceleration profile is performed by the subroutine PROFILE (see block diagram on fig. 3.1a,b,c.), and position reference is obtained from subroutine REFERENCE whose block diagram is given on fig. 3.2. Description of these subroutines is given in the following subchapters.

Now the value of control signal can be calculated by subroutine PIDFF (PID with feedforward).

When the required calculations are finished the control signals are output via D/A converters to the analog servocontrollers.

3.2. Subroutine PROFILE

This subroutine consists of three consecutive integrators. The input to this routine is the second derivative of maximal acceleration calculated earlier by main program. As the output the desired velocity is obtained.

Block diagram of this routine is presented on fig. 3.1a,b,c. Since very fast calculations are required, it is necessary to use only integer values during integration. Special precautions have been taken to avoid rounding errors and reduce the execution time. Therefore arithmetic shift right has been used instead of signed division. For this to be possible the sampling interval has been modified to the value $\Delta t = 0.977$ [ms] so that its inverse gives integer number which is simultaneously the number 10th power of 2. This means that by shifting 10 times to the right the same result is obtained as from division by 1/T.

The digits which are shifted out can not be neglected because this would introduce an accumulating error which would make the whole effort to control the acceleration profile unreasonable. To avoid this the sum of all round-ups is constantly kept in memory and the integration result is properly corrected whenever the total round-up becomes greater then 1.

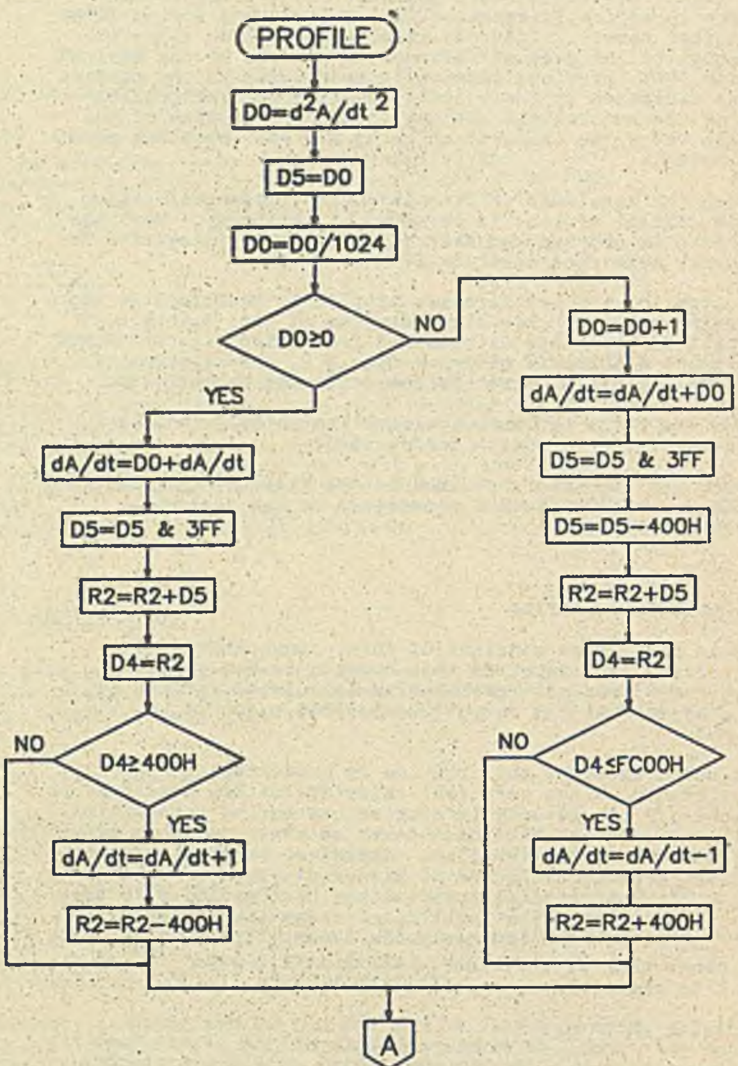
PROFILE
D0=d2A/dt 2
D5=D0
D0=D0/1024
D0≥0
NO
YES
D0=D0+1
dA/dt=dA/dt+D0
D5=D5 & 3FF
D5=D5-400H
R2=R2+D5
D4=R2
D4≤FC00H
NO
YES
dA/dt=dA/dt-1
R2=R2+400H
dA/dt=D0+dA/dt
D5=D5 & 3FF
R2=R2+D5
D4=R2
D4≥400H
NO
YES
dA/dt=dA/dt+1
R2=R2-400H
A

Fig. 3.1a.

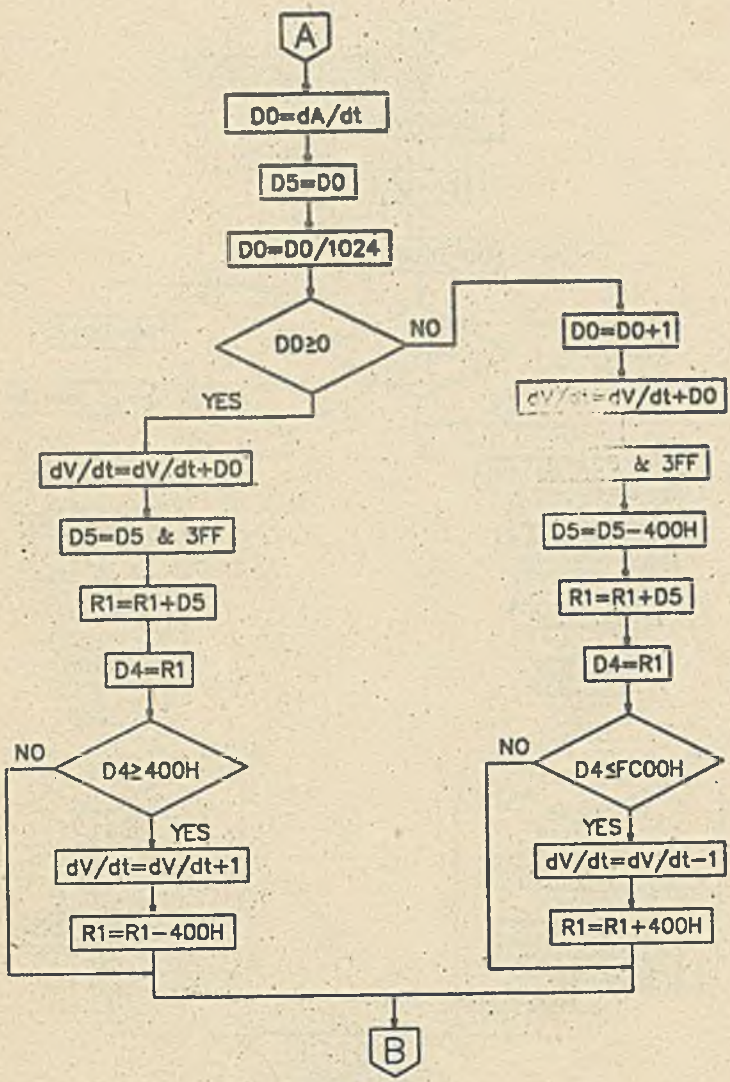
A
D0=dA/dt
D5=D0
D0=D0/1024
D0≥0
NO
YES
D0=D0+1
dV/dt=dV/dt+D0
& 3FF
D5=D5-400H
R1=R1+D5
D4=R1
NO
D4≤FC00H
YES
dV/dt=dV/dt-1
R1=R1+400H
dV/dt=dV/dt+D0
D5=D5 & 3FF
R1=R1+D5
D4=R1
NO
D4≥400H
YES
dV/dt=dV/dt+1
R1=R1-400H
B

Fig. 3.1b.

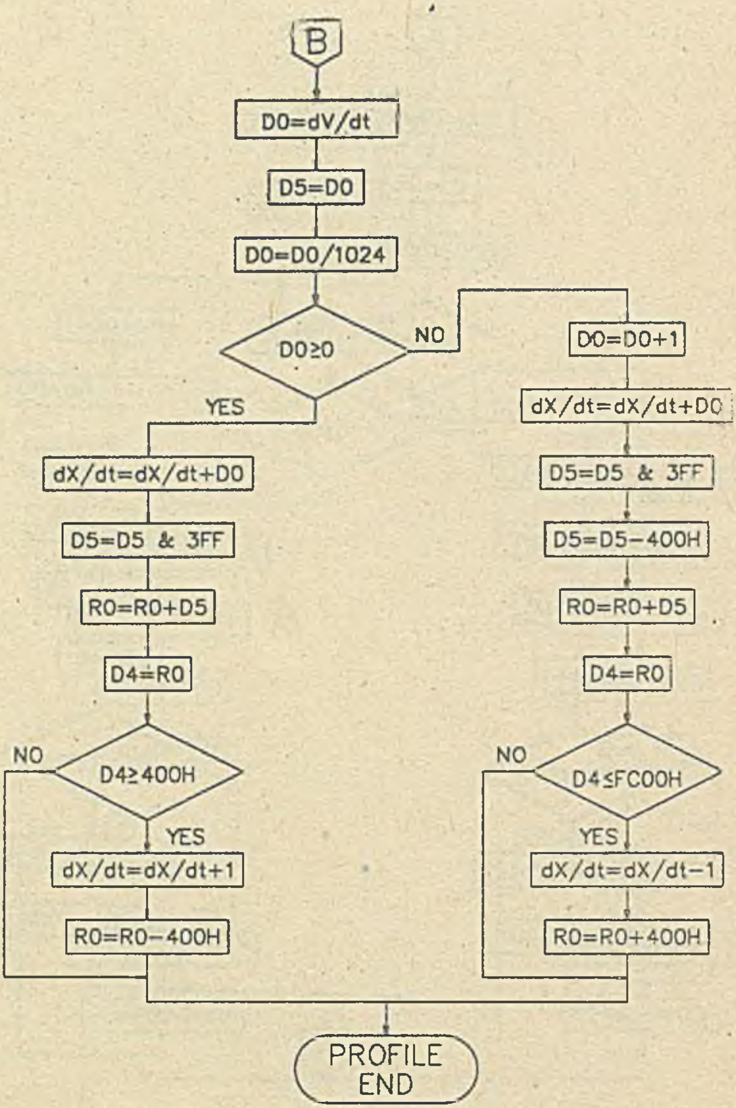

Fig. 3.1c.

Fig. 3.1. Net diagram of the subroutine PROFILE
Rys. 3.1. Schemat sieciowy procedury PROFILE

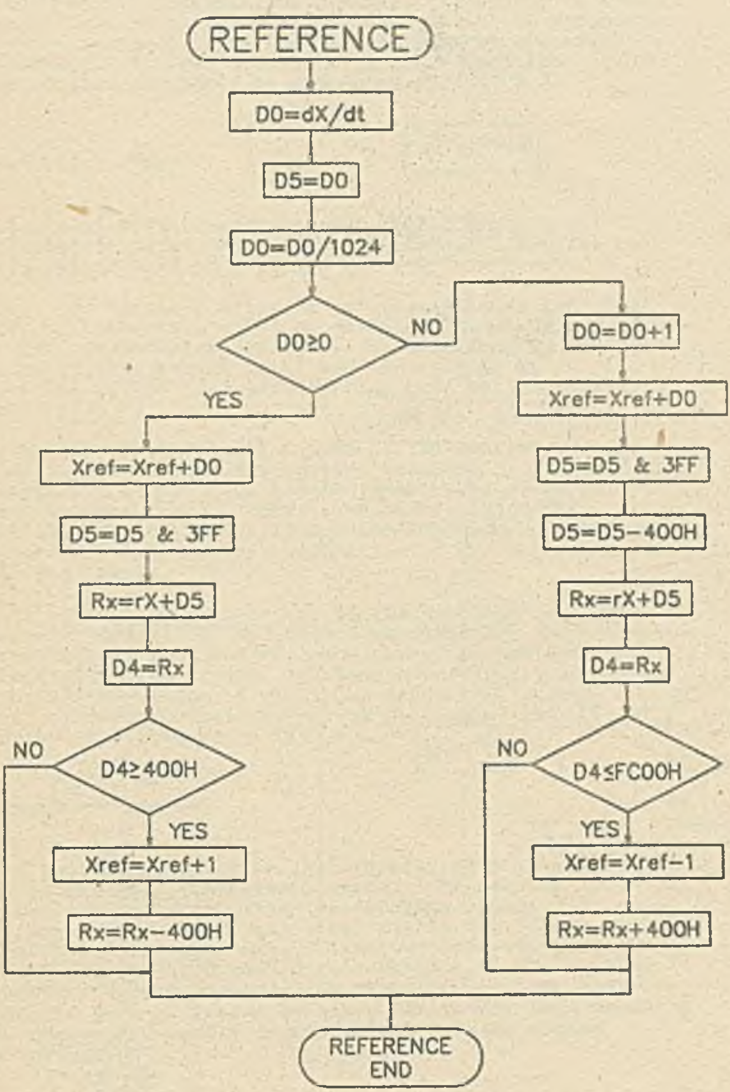

Fig. 3.2.

Fig. 3.2 Net diagram of the subroutine REFERENCE
Rys. 3.2 Schemat sieciowy procedury REFERENCE

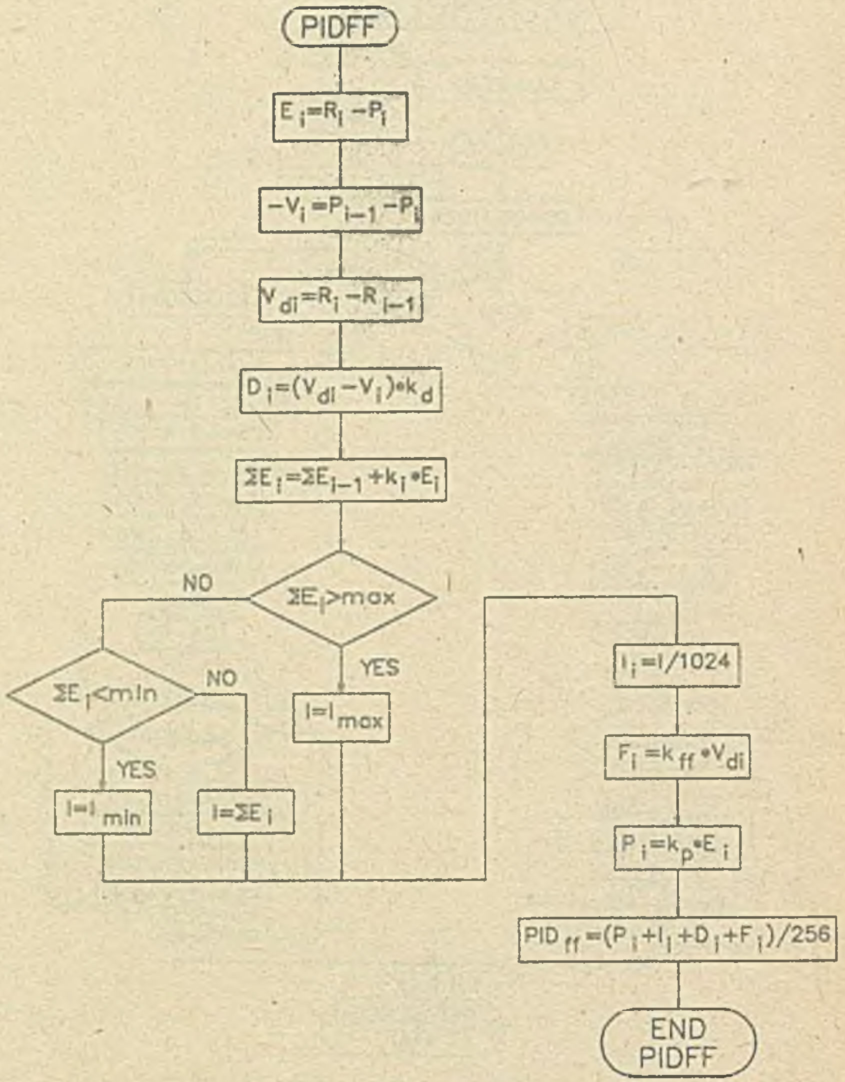

Fig. 3.3. BLOCK DIAGRAM OF PID CONTROLLER WITH FEEDFORWARD

Rys. 3.3 Schemat blokowy regulatora PID z kompensacją.

3.3. Subroutine REFERENCE

This routine is a single integrator identical to these used to integrate acceleration profile. The input to this routine is desired velocity and as the result the reference position is returned. The same precautions to guarantee high accuracy and speed are taken as above described. Block diagram of this procedure is presented on fig. 3.2.

3.4. Suboutine PIDFF

This routine is executed always during the interruptservicing process. The PID control algorithm with velocity feedforward is implemented here.

First the position error is calculated in D0 and saved in memory. Then minus velocity measured is calculated in D1 and saved. Next desired velocity is calculated in D3. After adding this result to D1 and multiplying by differential factor K_d a differential part of controller is obtained.

To obtain the integration part of the controller the error multiplied by the integration constant k_i is added to the integrated error value and the integrated error is shifted 10 times right. This gives the I part of controller. Top and bottom limits are imposed on the integrated error to prevent saturation of the integrator in case of big errors.

Multiplying the contents of D3 (D3 contains the reference velocity) by feedforward constant k_{ff} the feedforward part is obtained. The sum of proportional, differential, integrational and feedforward parts scaled by shifting 8 times right is the value of control signal returned by this subroutine. Block diagram of this subroutine is presented on fig. 3.3.

Conclusions

The controller which has been presented here has been tested on the real experimental setup. The positon errors of the rotational joints during the tracking and/or acceleration were always contained within the range <-1,+1> [mrad] for both axes. Higher accuracy is possible to obtain if the encoders with higher resolution are applied. No overshots were present during transients if the software PID and the gain of analog velocity controller were properly tuned. The static errors were also not present during acceleration.

Recenzent: Prof. dr hab. inż. Z. Bubnicki

Wpłynęło do Redakcji do 1988-04-30.

ŁĄCZNA SYNCHRONIZACJA RUCHU W ROBOTACH ŚLEDZĄCYCH

S t r e s z c z e n i e

Prezentowany artykuł jest raportem z badań dotyczących synchronizacji ruchu manipulatora za pomocą regulatorów zdolnych do nadążania za ruchomymi obiektami. Koncepcja harware'u do sterowania jest przedstawiona wraz z analizą spodziewanych statycznych i dynamicznych błędów śledzenia. Podkreśla się i głęboko analizuje ważność kompensacyjnego sprzężenia prędkościowego. Przedstawione również zostały wybrane schematy blokowe oprogramowania sterowania wykorzystującego dyskutowane koncepcje.

СУММАРНАЯ СИНХРОНИЗАЦИЯ ДВИЖЕНИЯ В СЛЕДЯЩИХ МАНИПУЛЯТОРАХ

Р е з ю м е

В работе представлены итоги исследований синхронизаций движения манипулятора с помощью регуляторов способных следить за подвижными объектами. Концепция управляющего хардвара берет во внимание статические и динамические ошибки слежки. Особенно важные являются компенсатённые скоростные связи. Представлено тоже избранные блоковые схемы опрограмирования управления использующего дискутированные концепции.