

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ V LIBERCI

Fakulta strojní

Katedra strojů průmyslové
dopravy

Školní rok: 1990/91

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro Martin A d a m e c

obor Spalovací motory

Vedoucí katedry Vám ve smyslu zákona č. 172/1990 Sb. o vysokých školách určuje tuto diplomovou práci:

Název tématu: Zpracování experimentálních výsledků
s využitím statistických metod a malé
výpočetní techniky

Zásady pro vypracování:

Proveďte rozbor možností stanovení závislostí parametrů vznětového motoru s využitím statistických metod a navrhnete řešení pro jednotlivé parametry motoru.

Vypracujte výpočtové programy pro zpracování výsledků zkoušek přeplňovaného motoru na počítači PC-AT včetně grafických výstupů sledovaných veličin.

V 9.7.1991

VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ
Ústřední knihovna
LIBEREC 1, STUDENTSKÁ 8
PSČ 461 17

KSD/SPD

Rozsah grafických prací: cca 30 stran průvodní zprávy
Rozsah průvodní zprávy: výpočtové programy (průběhy
Seznam odborné literatury: charakteristik motoru a vybraných provozních parametrů motoru)

KADEŘÁBEK, J.: Matematika III, VŠST, Liberec 1979

LINCZENI, A.: Inženýrská statistika. Alfa, Bratislava 1973

Vedoucí diplomové práce: Ing. Lubomír Moc, CSc.

Konzultant:

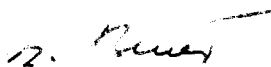
Zadání diplomové práce: 31. 10. 1990

Termín odevzdání diplomové práce: 3. 5. 1991




Doc. Ing. Stanislav Beroun, CSc.

Vedoucí katedry


Prof. Ing. Zdeněk Kovář, CSc.

Děkan

V Liberci

dne 31. 10. 1990

Vysoká škola strojní a textilní

Fakulta strojní

katedra strojů průmyslové dopravy

obor EM - MD - E

stroje a zařízení pro strojírenskou výrobu

zaměření

stroje a zařízení pro průmyslovou dopravu

ZPRACOVÁNÍ EXPERIMENTÁLNÍCH VÝSLEDKŮ S VYUŽITÍM STATISTICKÝCH METOD A MALÉ VÝPOČETNÍ TECHNIKY

Verdina ADAMEC

KSP - 211

Vedoucí práce : Ing. Ludomír KOC , VŠET Liberec

Název práce : Příloha

.....

Počet stran : 47

Počet tabulek : 0

Počet obrázků : 10

Počet příloh : 2

Vysoká škola strojní a textilní

fakulta strojní

katedra strojů průmyslové dopravy

obor 23 - 20 - 8

stroje a zařízení pro strojírenskou výrobu

zaměření

stroje a zařízení pro průmyslovou dopravu

ZPRACOVÁNÍ EXPERIMENTÁLNÍCH VÝSLEDKŮ S VYUŽITÍM STATISTICKÝCH METOD A MALÉ VÝPOČETNÍ TECHNIKY

Martin ADAMEC

KSD - 211

Vedoucí práce : Ing. Lubomír MOC , VŠST Liberec

Rozsah práce a příloh

Počet stran : 47

Počet tabulek : 0

Počet obrázků : 10

Počet příloh : 6

Místopřísežné prohlášení

Místopřísežně prohlašuji, že jsem diplomovou práci
vypracoval samostatně s použitím uvedené literatury.

V Liberci 25.4.1991

Colanec Martin

Poděkování

Děkuji svému vedoucímu diplomové práce Ing. Lubomíru Mocovi za cenné rady a připomínky, které mi pomohly zdárně zakončit tuto diplomovou práci.

OBSAH

1. Použitá označení
2. Úvod
3. Charakteristika motoru LIAZ M2-650
 - 3.1 Popis motoru
 - 3.2 Hlavní technické údaje motoru
4. Popis měřicího stanoviště
5. Charakteristiky motorů
 - 5.1 Provozní oblasti
 - 5.2 Vlivy na výkon a točivý moment
 - 5.3 Základní charakteristiky
 - 5.4 Zvláštní charakteristiky
6. Statistické metody vyhodnocování výsledků
 - 6.1 Pojem korelační závislosti
 - 6.2 Lineární regrese
 - 6.3 Metoda nejmenších čtverců
 - 6.4 Nelineární regrese
 - 6.5 Funkcionální regrese
 - 6.6 Korelační úloha
 - 6.7 Měření těsnosti lineární závislosti
 - 6.8 Měření těsnosti jednoduché lineární závislosti
7. Vyhodnocování naměřených veličin
8. Programy používané na vyhodnocování výsledků
9. Popis vytvořeného programu
10. Závěr
11. Použitá literatura

1. Použitá označení

n	[1/min]	-otáčky motoru
n_t	[1/min]	-otáčky turbodmyhadla
M_t	[Nm]	-točivý moment motoru
τ	[s]	-čas spotředy paliva
M	[g]	-množství spotřebovaného paliva
t_o	[°C]	-teplota oleje
t_w	[°C]	-teplota chladicí kapaliny
t_{1T1}	[°C]	-teplota výfukových plynů 1., 2. a 3. válce
t_{2T1}	[°C]	-teplota výfukových plynů 4., 5. a 6. válce
t_{T2}	[°C]	-teplota spalin před turbínou
$t_{T1..4}$	[°C]	-teplota výfukových plynů na 1. válci
t_{D1}	[°C]	-teplota vzduchu před dmyhadlem
t_{D2}	[°C]	-teplota vzduchu za dmyhadlem
t^*_{D3}	[°C]	-teplota plnicího vzduchu před mezichladičem
t_{D3}	[°C]	-teplota plnicího vzduchu za mezichladičem
t_p	[°C]	-teplota plnicího vzduchu do motoru
t_{w1}	[°C]	-teplota vody na vstupu do mezichladiče
t_{w2}	[°C]	-teplota vody na výstupu z mezichladiče
p_o	[MPa]	-tlak oleje
p_{PT1}	[kPa]	-přetlak spalin před turbínou
p_{PT2}	[kPa]	-přetlak spalin za turbínou
p_{PD1}	[kPa]	-přetlak vzduchu před dmyhadlem
p_{PD2}	[kPa]	-přetlak vzduchu za dmyhadlem
$p_{D2K in}$	[kPa]	-kinetická složka tlaku vzduchu
p_{D2ce1K}	[kPa]	-celkový tlak vzduchu za dmyhadlem
p_{pp}	[kPa]	-plnicí přetlak vzduchu
p_{pCH}	[kPa]	-tlaková ztráta na mezichladiči
p_p	[kPa]	-tlakový spád na clonce

k	[%]	-kouřivost
P_e	[kW]	-efektivní výkon
p_e	[MPa]	-střední efektivní tlak
m_{pe}	[g/kWh]	-měrná spotřeba
m_{px}	[g/cy]	-měrná spotřeba na cykl
m'_{px}	[g/cy.1]	-měrná spotřeba na cykl
ρ_{vo}	[kg/m ³]	-hustota nasávaného vzduchu
ρ_{vp1}	[kg/m ³]	-hustota plnicího vzduchu
λ	[-]	-součinitel přebytku vzduchu
π	[-]	-stlačení dmyhadla
i	[-]	-počet válců
V_m	[dm]	-zdvihový objem motoru
M_b	[kg/s]	-spotřeba paliva
M_v	[kg/s]	-spotřeba vzduchu
y_i		-závisle proměnná veličina
x_i		-nezávisle proměnná veličina
y'		-teoretická hodnota závisle proměnné veličiny
a_{yx}		-konstanta regresní rovnice
b_{yx}		-regresní koeficient
s_x		-rozptyl nezávisle proměnné
s_{yx}		-kovariance y na x, reziduální rozptyl
n		-rozsah souboru
$s_{y'}$		-rozptyl odhadů y'
s_y		-celkový rozptyl závisle proměnné y
r_{yx}		-koeficient korelace
r_{yx}^2		-koeficient determinace
I_{yx}		-index korelace
I_{yx}^2		-index determinace

2. ÚVOD

Naftové motory se uplatňují ve všech oblastech hospodářství jako důležitý zdroj energie. Od jejich vzniku jsou hledány cesty zvyšování jejich výkonu, technické dokonalosti a zlepšování hospodárnosti provozu. Nejvýznamnější z nich je přeplňování motoru, které se používalo již v devadesátých letech minulého století, ale maximální využití našlo až po 2. světové válce zdokonalením turbodmychadel.

Typickým vývojovým trendem je zvyšování měrného výkonu při současném snižování hmotnosti. Tomuto směru právě nejlépe vyhovují přeplňované motory.

Cesta vývoje vedla přes nízkotlaké přeplňování, kde byl přírůstek výkonu nízký, až po vysokotlaké přeplňování, kde výkon motoru vzroste o více jak 50% výkonu motorů nepřeplňovaných.

Proto je pro výzkum a vývoj nutné zvládnout intenzivní a komplexní vývoj a zkoušky motorů s vysokotlakým přeplňováním.

Dosavadní vývoj motorů je zaměřen především na soustavné zvyšování středního efektivního tlaku, střední pístové rychlosti, zvyšování hospodárnosti a spolehlivosti, snižování škodlivých emisí ve formě spalin i hluku.

Pro všechny tyto úkoly je nutné experimentální ověření provedených úprav motoru. Je nutné přesně změřit parametry motoru a správně je vyhodnotit, abychom mohli porovnat parametry nových motorů s motory dosavadními, nebo s motory konkurenčními.

Nejjednodušším zpracováním výsledků je jejich výpočet a zápis do protokolu nebo tabulky. Toto je však nepřehledné. Mnohem lepší pro představu vývoje jednotlivých veličin je zpracování do grafů. Tyto grafy se nazývají charakteristiky motoru. Můžeme je kreslit různými způsoby. Naměřené body můžeme spojit přímo, ale tento způsob kreslení se obvykle nepoužívá. Nejvhodnější je proložit danými body křivku, a to buď tzv. od oka (to je nepřesné), nebo použijeme matematicko-statistické zpracování výsledků. Tato metoda se nazývá regresní analýza a při své práci využívá korelační počet. Moje diplomová práce se zabývá právě vyhodnocením výsledků pomocí regresní analýzy. Mojí úkol je

vyhodnotit zkoušky motoru a vypracovat program na zpracování těchto výsledků a na jejich výpis ve formě grafů nebo protokolu o zkoušce.

3.Charakteristika motoru LIAZ M2-650

3.1 Popis motor

Motor LIAZ M2-650 je naftový, čtyřdobý s přímým vstřikem paliva, s rozvodem ventilů OHV, pravotočivý, přepřínávaný turbodmychadlem poháněným výfukovými plyny, provedení motoru je stojaté s šesti válci v jedné řadě. Plnicí vzduch přicházející od turbodmychadla k hlavám válců je ochlazován mezichladičem, který je na automobilu namontován před chladičem vody. Chladicí soustava je kapalinová, přetlaková s uzavřeným oběhem a tvoří ji vodní čerpadlo, ventilátor termoregulatory, chladič, expansní nádržka a přestavitelná clona chladiče. Mazání motoru je tlakové, obstarává ho zdvojené zubové čerpadlo s přečerpávací a tlakovou sekcí. Filtraci mazacího oleje zajišťují dva paralelní plnoprátočné čističe a odstředivý čistič oleje. Mazací okruh je vybaven výměníkem tepla lamelového typu, průtok výměníkem je řízen v závislosti na pracovní teplotě mazacího oleje pomocí termoregulačního ventilu. Z okruhu mazání motoru je tlakově mazané i turbodmychadlo, kompresor a vstřikovací čerpadlo. Vstřikovací souprava je buď BOSCH nebo MOTORPAL s nezbytnými čističi paliva-hrubým čističem paliva na dopravním čerpadle a dvoustupňovým čističem s hrubou a jemnou čistící vložkou. Klikový hřídel je výkovek z oceli, v bloku je uložen v sedmi kluzných ložiskách. Písty z hliníkové slitiny se zalitým nosičem 1. pístního kroužku jsou osazeny třemi pístními kroužky a jsou chlazeny postřikem mazacího oleje. Hlava válce se 2

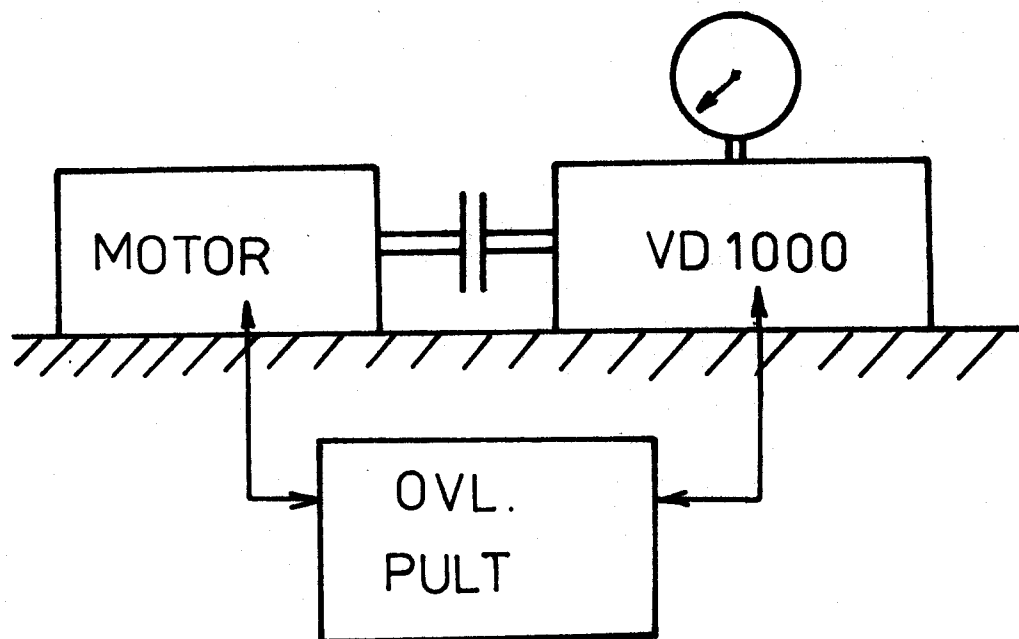
sacími a 2 výfukovými ventily je pro každý válec samostatná. Vodní čerpadlo je poháněno ozubeným kolem zevnitř z čela motoru. Odvětrání vnitřních prostorů motoru je provedeno hrdlem s odlučovačem oleje namontovaném na zadním víku. Motor má nezbytné doplňující příslušenství, t.j. alternátor pro dobíjení akumulátorů, kompresor pro plnění vzduchu do brzdové soustavy vozidla a zubové čerpadlo pro dodávku tlakového oleje pro servořízení.

3.2 Hlavní technické údaje motoru

Typ motoru	M2-650
Čistý výkon	
/ČSN 30 2008/	270 kW $\pm$ 5% při 2000 ot/min
Točivý moment Mt max.	1550 Nm $\pm$ 5% při 1300 ot/min
Měrná spotřeba paliva	
minimální	206 g/kWh
Palivo-motorová nafta	NM4, NM22, NM35
Způsob vstřikování paliva	přímý
Pracovní oběh	čtyřdobý
Počet válců	6
Uspořádání válců	stojaté, jednořadové
Rozvod	OHV
Plnění válců	turbobmychadlem
Smysl otáčení	pravotočivý
Způsob chlazení	kapalinový, přetlakový
Vrtání válců	135 mm
Zdvih pístů	160 mm
Objem válců	13.741 dm ³
Kompresní poměr	15:1
Poradí vstřiků	1-5-3-6-2-4
Předstih vstřiku	29° před HÚ
Nejvyšší užitečné otáčky	2000 ot/min
Max. předěhové otáčky	2340 ot/min

Volnoběžné otáčky	600 až 650 ot/min
Mazání motoru	
Provedení	tlakové, zubovým čerpadlem
Výkon čerpadla	8100 l/hod při 2000 ot/min motoru, protitlaku 540 kPa a teplotě oleje 85°C
Tlak mazacího oleje v provozních otáčkách	500 kPa
Tlak mazacího oleje za volnoběhu	min 100 kPa
Chlazení motoru	
Druh chlazení	kapalinové, přetlakové
Výkon čerpadla	406 l/min při 2000 ot/min motoru a tlak. spádu 200 kPa
Začátek otevírání termo- regulátorů	76± 2°C
Úplné otevření termoregu- látorů	max. 90°C
Příslušenství motoru	
Zubové čerpadlo	U 16.R.06
Turbodmychadlo ČZM	K36 3770 V/25.22
Kompresor	Typ 4136

2. Popis měřicího zařízení



obr. 1

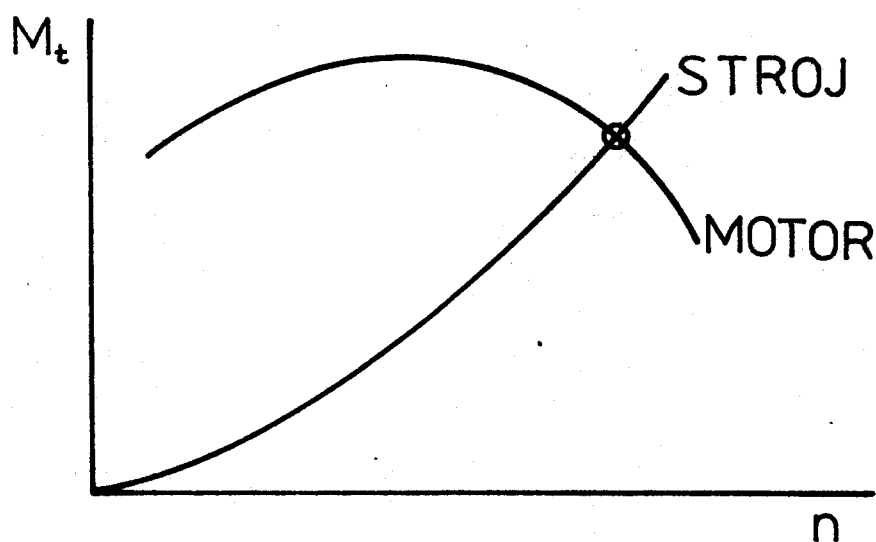
Motor M2.4R je při našich měřeních ustaven na měřicím stanovišti. Toto se skládá z dynamometru VD 1000, který je pružnou spojkou připevněn k měřenému motoru, ovládacího pultu a měřících přístrojů. Točivý moment se odečítá na ukazateli váhy dynamometru. Otáčky motoru se měří elektronicky deviačním otáčkoměrem EV 85. Spotřeba paliva se měří hmotnostním způsobem, stanovením doby, za jakou motor spotřebuje dané množství paliva. K tomu se používá odměrná nádoba, váha, sada závaží a stopky. Teplota výfukových plynů se měří termočlánkem Fe-Co, ostatní teploty se měří odporovými teploměry Pt 100. Tlaky p_0 a p_1 se měří a zobrazují na kruhovém deformačním manometru. Tlaky p_{p1} , p_{p2} , p_{p3} , p_{p4} se měří pomocí U-manometru s náplní vody. Tlaky p_{p5} a p_{p6} se měří U-manometrem s náplní rtuti. Kouřivost motoru se měří filtračním způsobem snímačem ÚVMV. Obr. 1 znázorňuje základní schema měřicího stanoviště. Pro některá speciální měření se používají další přídatné přístroje a speciální přípravky. Popsané měřicí zařízení je však stan-

darčí, a je používáno ve velké většině měření. Možné obměny se většinou týkají způsobu měření jednotlivých veličin a jeho přesnosti.

5. Charakteristiky motorů

5.1 Provozní oblasti motorů

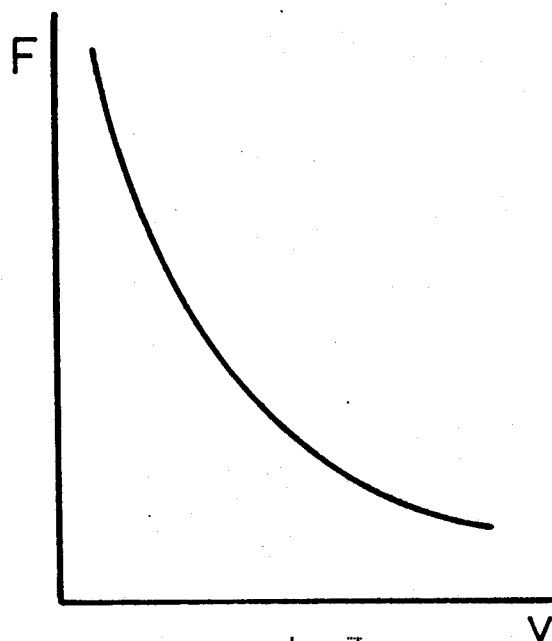
Pohání-li spalovací motor nějaký stroj, bude za všech okolností počet otáček obou strojů vždy takový, aby se výkon motoru rovnal příkonu stroje i s převodovými ztrátami. Sestrojí-li se momentová charakteristika motoru a poháněného stroje ve stejných souřadnicích, pak bude počet otáček v průsečíku příslušných charakteristik (viz obr.2).



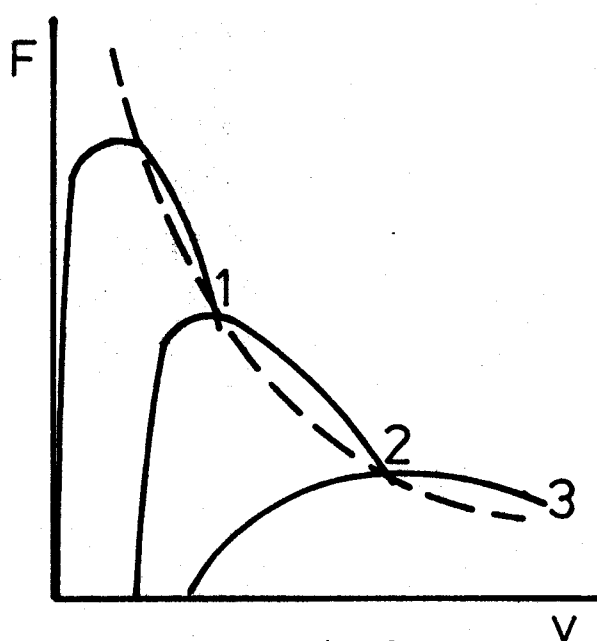
obr.2

Závislost hnací síly F na rychlosti vozu v je pro plynulou převodovku rovnocná hyperbola. Tato charakteristika je nejvýhodnější, protože při každé hnací síle pracuje motor s

neproměnným, konstantním počtem otáček, a při stálé měrné spotřebě paliva (viz obr.3).



obr.3



obr.4

Další obrázek (obr.3) znázorňuje motor s převodovkou s konečným počtem převodových stupňů. Provozní rozmezí mezi jednotlivými stupni musí překlenout motor, který tím do jisté míry přejímá funkci převodovky. Výkon motoru má být v tomto rozmezí konstantní, aby se dosáhlo hyperbolického průběhu hnací síly F . Čím širší je rozmezí otáček, v němž je výkon motoru neproměnný, tím menší je potřebný počet převodových stupňů.

Z požadavku na řazení vyplývají požadavky na vozidlový motor. V rozmezí používaného počtu otáček má motor vykazovat co nejehospodárnější provoz. Momentová charakteristika má být v oblasti blízké maximálnímu momentu plochá, a výkon motoru mezi převodovými stupni má být neproměnný, to znamená, že motor má být pružný.

Požadavky na jiné motory (stacionární, lodní, letadlové, ...) jsou jiné a to podle daných specifických podmínek.

Charakteristika dává větší střední efektivní tlak p_e než křivka 3 (obr.4). Přísluší pružnému motoru, který nevyžaduje

tak časté řazení. Tato křivka dává největší střední efektivní tlak při vysokých otáčkách. Výkon motoru je větší, ale stabilita menší. Takto seřízený motor přejímá funkci převodovky mezi jednotlivými převodovými stupni jen v úzkém rozsahu otáček, což je pro vozidlové motory nevýhodné. Takový motor se nazývá nepružný.

5.2 Vlivy na výkon a točivý moment

Výkon motoru závisí na mnoha činitelích a nedá se předem přesně stanovit. Základní vlivy lze určit ze vzorce, který se používá pro stanovení hlavních rozměrů motoru (1).

$$P_e = \frac{i \cdot V_z \cdot n \cdot p_e}{c} \quad (1)$$

kde P_e je efektivní výkon motoru, i je počet válců, V_z je zdvihový objem jednoho válce, p_e je střední efektivní tlak a c je konstanta různá pro dvoudobé a čtyřdobé motory. Přirozená cesta pro zvyšování výkonu je zvyšování zdvihového objemu motoru a počtu válců. Největší možnosti jsou však ve zvyšování součinu $n \cdot p_e$. Vzrůst tlaku p_e udává hlavní směr pro zvyšování točivého momentu motoru a tím i jeho výkonu. Dalším směrem je zvyšování provozních otáček motoru, což se však nepříznivě projevuje poklesem životnosti motoru.

5.3 Základní charakteristiky

Charakteristiky motoru znázorňují závislost charakteristických veličin motoru (např. výkonu, točivého momentu, měrné spotřeby, ...) na jednom z parametrů, jenž může být měřítkem chodu (počtu otáček, ...).

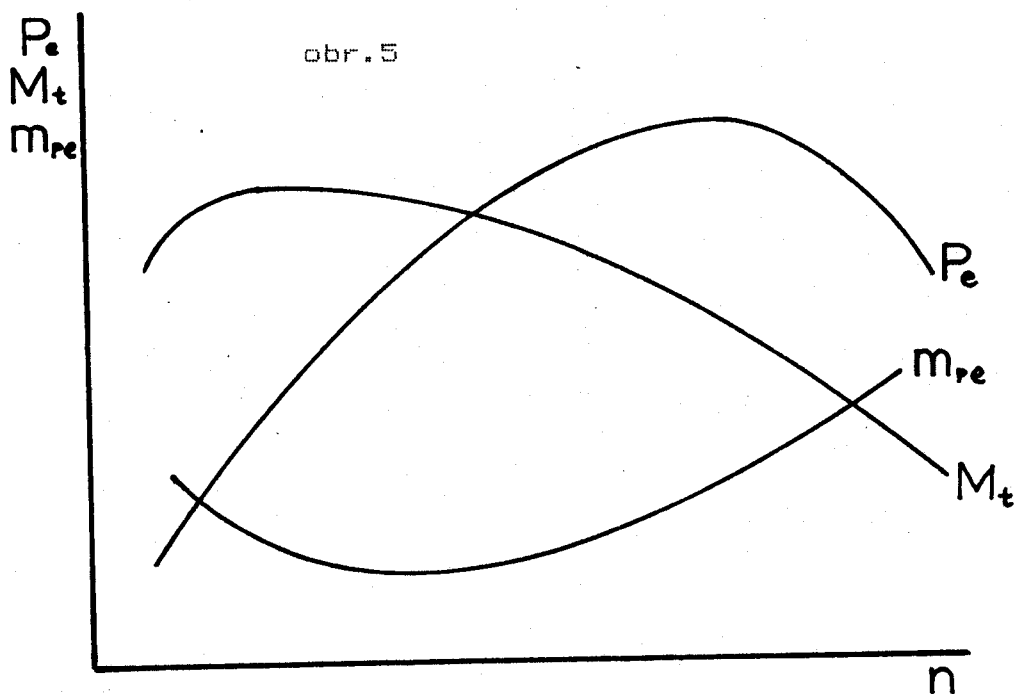
V této souvislosti se setkáváme s celou řadou charakteristik. Můžeme kreslit charakteristiky např. vlivu seřízení předstihu vstřiku a zážehu, vliv parametrů nasávaného vzduchu nebo směsi, seřízení karburátoru, atd.

Základní charakteristiky jsou však:

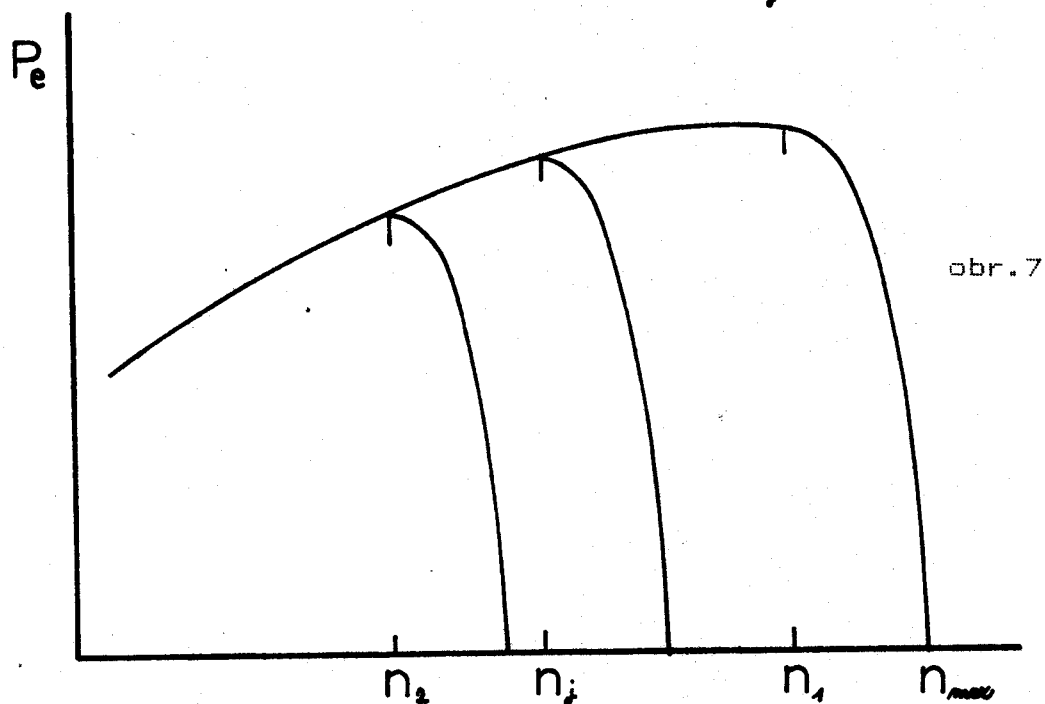
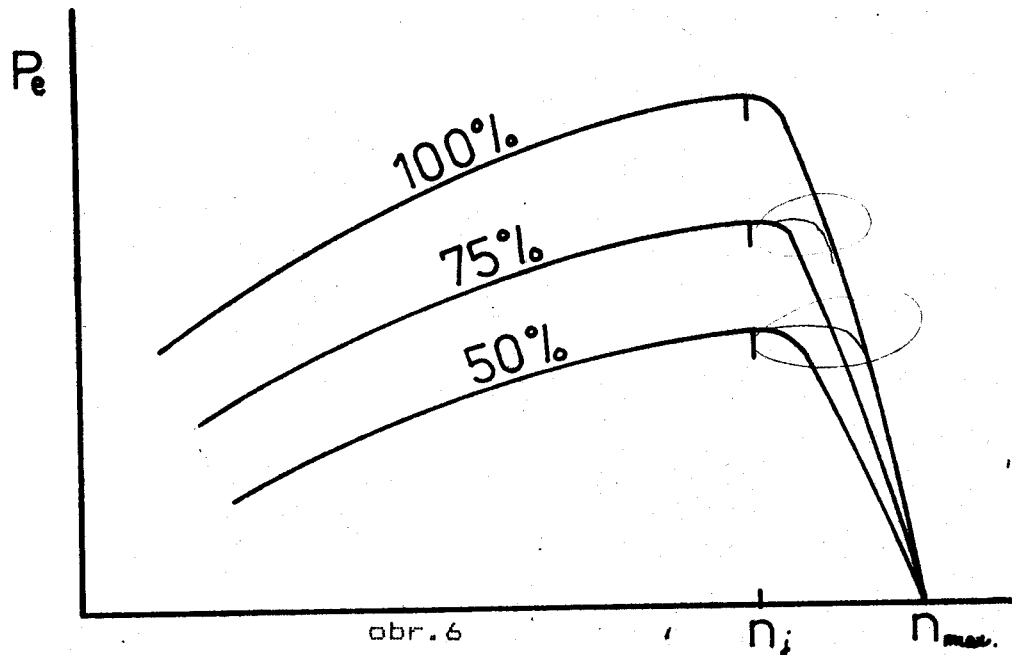
- otáčková (rychlostní) charakteristika
- zatěžovací charakteristika
- seřizovací charakteristika
- celková charakteristika

A. Rychlostní (otáčková) charakteristika

Udává závislost výkonu, točivého momentu, hodinové a měrné spotřeby a dalších veličin na otáčkách motoru. Stanoví se zatěžováním motoru při neproměnné poloze škrticí klapky nebo regulační tyče. Ostatní parametry jsou nastaveny na optimální hodnoty (předstih, předvsřik, teplota chladící kapaliny,...). Při plném otevření škrticí klapky karburátoru, nebo plném vysunutí regulační tyče vstřikovacího čerpadla dostaneme tzv. vnější rychlostní charakteristiku, při jiné poloze tzv. částečnou rychlostní charakteristiku. Na obr. 5 je otáčková charakteristika benzínového motoru.

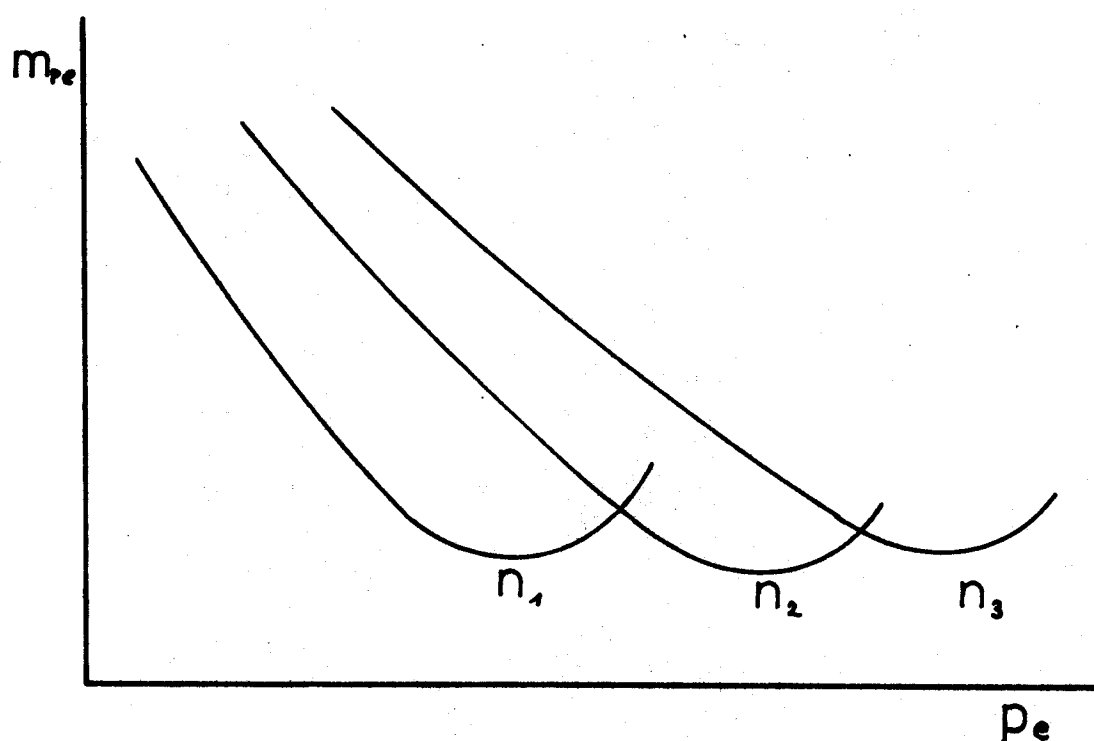


Stanovení rychlostní charakteristiky je dáno např. ČSN 09 0851 nebo ČSN 30 2008. Kromě vnějších a částečných charakteristik jsou používány např. charakteristika na mezi kouření, charakteristika běhu naprázdno, charakteristika s regulátorem nebo bez regulátoru. Na obr. 6 je uvedena charakteristika s omezovacím regulátorem, na obr. 7 je charakteristika s výkonostním regulátorem.



B. Zatěžovací charakteristika

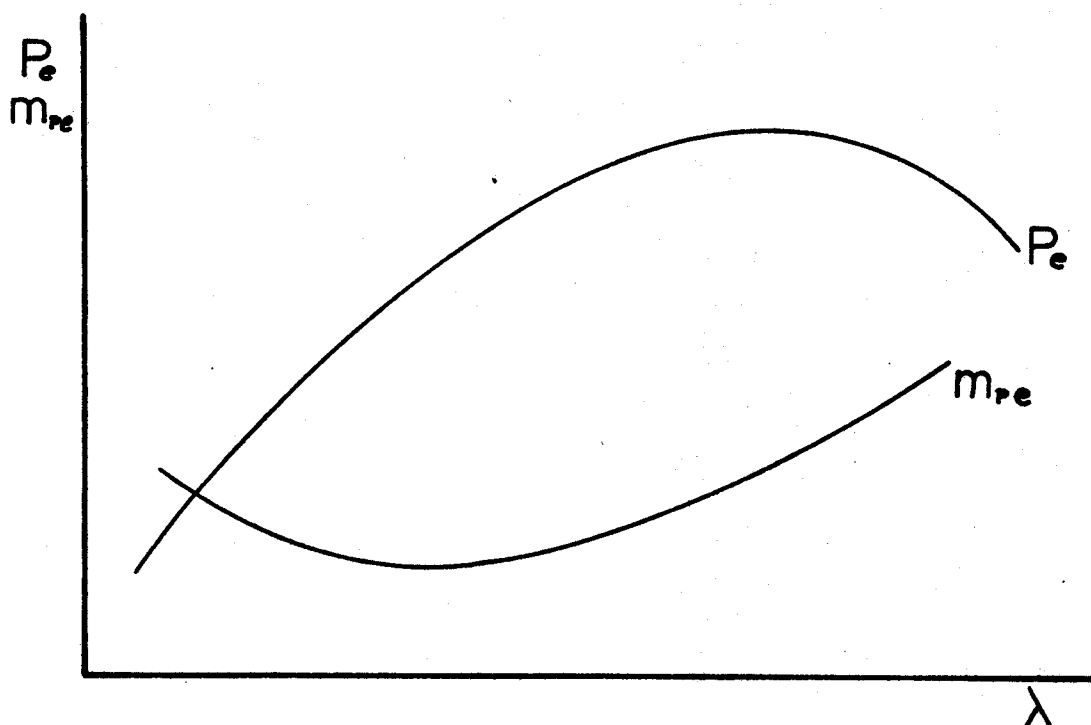
Znázorňuje závislost provozních veličin na některé veličině charakterizující zatížení motoru při stálých otáčkách. U naftových motorů bývá omezena hranicí tzv. kouřového chodu. Při nedokonalém hoření se protahuje spalování do expanzního zdvihu, zvyšuje se teplota stěn a pístu, které se přehřívají, tvoří se tuhé usazeniny na stěnách spalovacího prostoru, a volný, nespálený uhlík vychází s výfukovými plyny, které nabývají šedého až černého zbarvení. Na obr. 8 je příklad charakteristiky pro měrnou spotřebu.



obr. 8

C. Seřizovací charakteristika

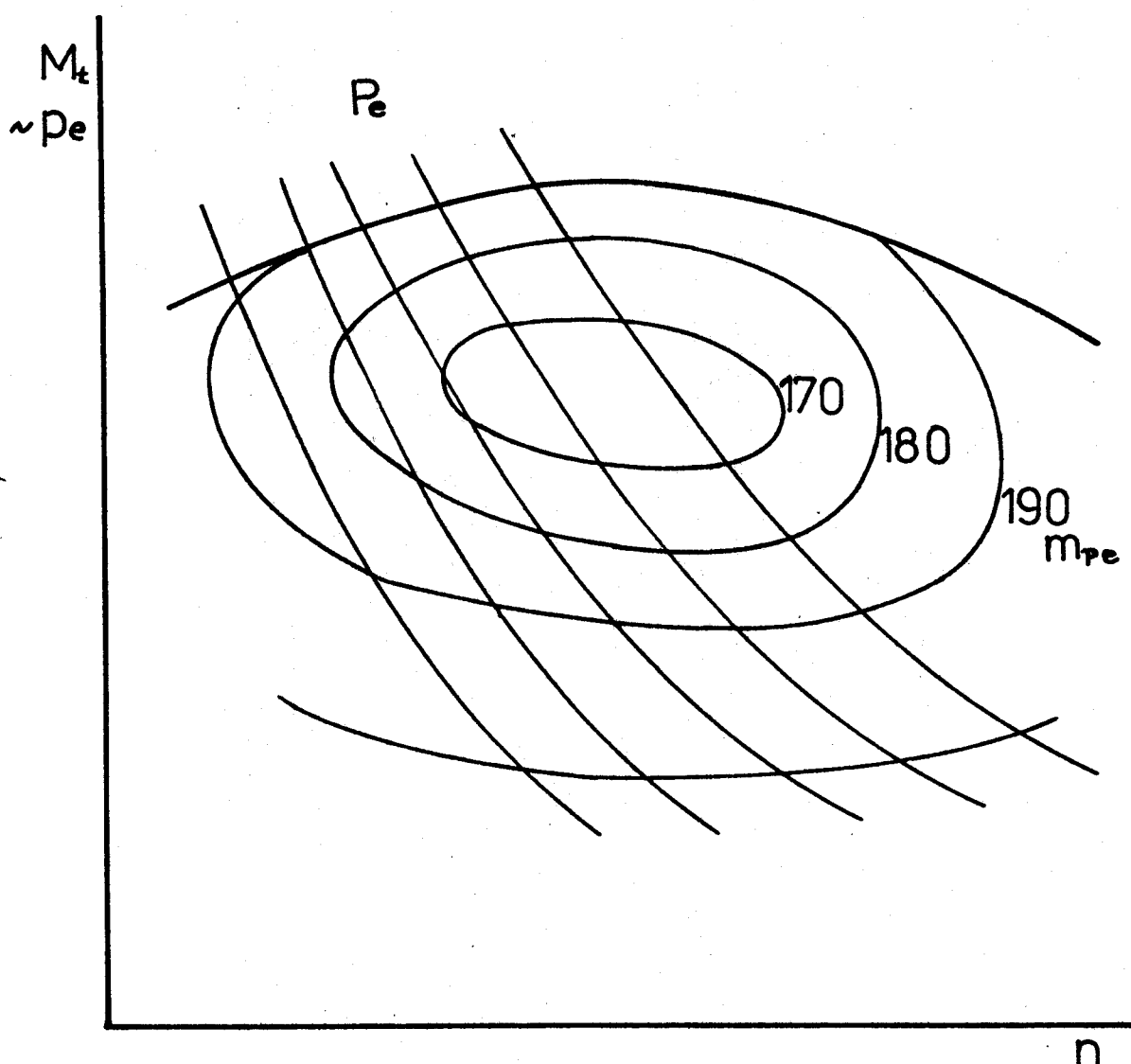
Znázorňuje závislost užitečného výkonu popř. dalších provozních veličin (měrné spotřeby paliva,...) na některé veličině charakterizující seřízení motoru. Na obr. 9 je uvedena seřizovací charakteristika benzínového motoru, ve které je seřizovanou veličinou součinitel přebytku vzduchu.



obr. 9

D. Celková charakteristika

Vyjadřuje závislost sledované provozní veličiny na dvou základních veličinách. Přitom každé křivce diagramu přísluší určitá stálá hodnota sledované veličiny jako parametr. Tato charakteristika se neměří, ale sestavuje se ze soustavy otáčkových a zatěžovacích charakteristik pomocí řezů. Na obr.10 je příklad celkové charakteristiky.



obr.10

E. Zvláštní charakteristiky

Jsou to charakteristiky, které byly naměřeny pro různé speciální stroje, nebo pro zvláštní režimy. Patří sem např. otáčková charakteristika lodní vrtule, výšková charakteristika, charakteristika přechodová, charakteristika závislosti na libovolném parametru,...

6. Matematická statistika

6.1. Pojem - korelační závislost

Při sledování souvislosti jevů a věcí se setkáváme s různými formami souvislostí. V podstatě rozeznáváme souvislosti vnější - nepodstatné a souvislosti vnitřní - podstatné. Podstatné souvislosti jsou takové, při kterých je vnější souvislost výrazem určité vnitřní nevyhnutelnosti. Charakteristickou formou příčinné závislosti je příčinná, neboli kauzální závislost, t. j. závislost, kdy jeden z jevů (příčina) vyvolává za určitých podmínek existenci druhého jevu - následku. Zkoumání příčinných závislostí je jedna z nejdůležitějších úloh vědy. Se zkoumáním a vyjadřováním závislosti mezi dvěma nebo více měnícími se veličinami, se setkáváme skoro v každé vědní disciplíně. Příčinná závislost se zpravidla projevuje v různých formách. Jednou z forem je funkční závislost, s kterou se setkáváme v matematické analýze. Podstata funkční závislosti je v tom, že každé hodnotě jedné proměnné veličiny, která se nazývá argument a značí se x (veličina nezávisle proměnná), se přiřazuje vždy určitá hodnota druhé proměnné veličiny, která se nazývá funkcí a značí se y (závisle proměnná). Funkční závislost je charakteristická především pro přírodní jevy - hlavně pro jevy neživé přírody.

Korelační závislost je taková statistická závislost, při které změnám hodnot jedné veličiny odpovídá změna průměrného rozložení četnosti druhé veličiny. Zkoumání korelační závislosti předpokládá zkoumání rozdělení veličiny y pro každé x a změny tohoto rozdělení se změnami x .

Korelační závislost je tedy výrazem volného příčinného vztahu, avšak funkční závislost vyjadřuje pevný příčinný vztah, na který nemá vliv náhoda, a kde příčině odpovídá bezprostředně účinek.

Funkční závislost je možné považovat za hraniční případ korelační závislosti, který by mohl nastat při vyloučení všech náhodných vlivů. Pak by ale nešlo o statistickou závislost.

Pojem statistické závislosti je tedy širší než pojem korelační závislosti. Pojem statistická závislost zahrnuje kromě korelační závislosti i zkoumání rozptylu, asymetrie, a pod. Korelační závislost veličin je možné zkoumat jen na základě celých souborů hodnot těchto veličin. Jen tak objevíme to, co je pro danou závislost v rámci existujících podmínek typické, a můžeme odhalit závislost, která umožňuje určit pravděpodobnou hodnotu jedné veličiny (závisle proměnné) na základě známé druhé veličiny (nezávisle proměnné).

Způsoby které používáme na měření korelační závislosti nazýváme korelační počet.

Úlohou korelačního počtu je:

- a) Zjistit formu závislosti a vyjádřit ji matematickou funkcí tzv. regresní křivkou, která umožňuje ze známých hodnot nezávisle proměnné veličiny určit velikost závisle proměnné veličiny = regresní úloha.
- b) Určit stupeň síly, s jakou se daná závislost projevuje uprostřed rušících vedlejších faktorů, v případě posoudit vhodnost vystihnoutí formy závislosti určitou funkcí = korelační úloha.

Korelace můžeme dělit podle různých kritérií takto:

- a) Jestliže zkoumáme závislost dvou veličin, hovoříme o jed-

noduché korelaci, jakmile zkoumáme závislost více jak dvou veličin, hovoříme o mnohonásobné korelaci.

b) Podle toho, jakým typem funkce je možné vystihnout formu dané závislosti, rozeznáváme opět korelaci lineární a nelineární. O lineární korelaci hovoříme tehdy, jakmile jsou změny hodnot jedné veličiny zhruba lineárně závislé na změně hodnot druhé veličiny, a když je možné vystihnout formu závislosti lineární funkcí dvou, nebo více proměnných. Ve všech ostatních případech hovoříme o nelineární korelaci.

c) Jestliže růst (pokles) hodnot jedné veličiny má za následek růst (pokles) hodnot druhé veličiny, pak hovoříme o přímé (pozitivní) korelaci. V případě, že růst hodnot jedné veličiny má za následek pokles hodnot druhé veličiny, a naopak, pak hovoříme o nepřímé (negativní) korelaci.

6.2. Lineární regrese dvou korelovaných veličin

Je to první úloha korelačního počtu při párové korelační závislosti, t.j. vyjádření formy závislosti dvou korelovaných veličin matematickou funkcí.

Na řadu podmíněných průměrů se můžeme dívat jako na první vyjádření regrese, která je však závislá na všech náhodnostech jednotlivých pozorování. Hovoříme o ní jako o empirické regresi. Podmínečné průměry neřeší vlastní regresní úlohu, protože jsou ovlivněny náhodnostmi jednotlivých jevů. Regresi proto vystihujeme tzv. regresní čarou, která vyjadřuje jen hlavní tendenci průběhu hodnot závisle proměnné, a je indiferentní vůči jejich zvláštnostem.

Nejjednodušší typ regrese dvou korelovaných veličin je lineární regrese. Hovoříme o ní tehdy, když můžeme korelační závislost dvou veličin vyjádřit lineární funkcí. Používáme ji často i při nelineárních průbězích, zvláště tehdy, když neznáme skutečnou formu korelační závislosti. V mnoha případech lze vhodnou transformací proměnných převést nelineární vztah

na lineární. Proto je lineární regrese nejvýznamnějším případem v oblasti ekonomického i technického zkoumání. Má také teoretický význam z hlediska zkoumání korelační závislosti. Při lineární regresi jde o vyjádření průběhu korelační závislosti regresní přímkou, pomocí které by bylo možné odhadovat z velikosti nezávisle proměnné x velikost proměnné y . Rovnici regresní přímky vyjádříme v tvaru

$$y' = a_{yx} + b_{yx} * x \quad (2)$$

kde symbol y' značí teoretické (vyrovnané) hodnoty, a ne empirické (naměřené) hodnoty závisle proměnné, které budeme i dále označovat y ; a_{yx} a b_{yx} jsou konstanty rovnice, které nazýváme regresní konstanty. Z matematického hlediska značí b_{yx} směrnici regresní přímky. V korelačním počtu nazýváme tuto konstantu regresním koeficientem. Vyjadřuje vlastní smysl regrese; ukazuje, jak se průměrně mění závisle proměnná veličina při jednotkové změně nezávisle proměnné veličiny. Je-li $b_{yx}=0$, proměnné y a x jsou korelačně nezávislé.

6.3. Metoda nejmenších čtverců

Na určení nejvýhodnější a nejsprávnější přímky pro účel odhadu jsou třeba objektivní kritéria. Existuje několik způsobů na analytické vyrovnání empirických údajů. Převládající postavení mezi nimi má metoda nejmenších čtverců.

Určení rovnice regresní přímky v podstatě znamená, že na místo skutečně naměřených hodnot závisle proměnné přiřadíme ke každé hodnotě nezávisle proměnné jakousi teoretickou hodnotu závisle proměnné (y_i'), která leží na přímce. Nejvhodnější přímka bude ta, při které je součet odchylek empirických hodnot od teoretických minimální, to znamená, že platí podmínka

$$S = \sum (y_i - y_i') = \min. \quad (3)$$

Pro lineární regresi-regresní přímku to znamená, že platí

$$S = \sum (y_i - a_{yx} - b_{yx} * x_i) = \min. \quad (4)$$

Ještěli vyžadujeme plnění všeobecné podmínky, aby součet čtverců odchylek empirických hodnot od teoretických byl minimální, pak platí

$$S = \sum (y_i - y_i')^2 = \min. \quad (5)$$

a také

$$S = \sum (y_i - a_{yx} - b_{yx} * x_i)^2 = \min. \quad (6)$$

potom hovoříme o tom, že regresní přímku určujeme metodou nejmenších čtverců. Jednoznačné určení regresní přímky tkví vlastně v jednoznačném určení regresních ukazovatelů a_{yx} a b_{yx} , které považujeme za neznámé konstanty. Výraz na levé straně rovnice je možno považovat za funkci hledaných parametrů přímky $F(a_{yx}, b_{yx})$ a celou úlohu můžeme formulovat jako hledání minima funkce dvou proměnných. Vycházíme tedy z požadavku, aby

$$\sum (y_i - y_i')^2 = \sum (y_i - a_{yx} - b_{yx} * x_i)^2 = F(a_{yx}, b_{yx}) = \min. \quad (7)$$

Z nutné podmínky pro minimum funkce dvou proměnných vyplývá, že první parciální derivace funkce $F(a_{yx}, b_{yx})$ podle dvou proměnných se musí rovnat nule. Z toho získáme dvě rovnice pro řešení parametrů a_{yx} a b_{yx}

$$\frac{\partial F(a_{yx}, b_{yx})}{\partial a_{yx}} = 2 \sum (y_i - a_{yx} - b_{yx} * x_i) * (-1) = 0 \quad (8)$$

$$\frac{\alpha F(a_{yx}, b_{yx})}{\alpha b_{yx}} = 2 \sum (y_i - a_{yx} - b_{yx} * x_i) * (-x) = 0 \quad (9)$$

a jejich úpravou získáme tzv. normální rovnice

$$\sum y_i = n * a_{yx} + b_{yx} * \sum x_i \quad (10)$$

$$\sum (x_i * y_i) = a_{yx} * \sum x_i + b_{yx} * \sum x_i^2 \quad (11)$$

a jejich řešením získáme vztahy pro

$$b_{yx} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (12)$$

a pro

$$a_{yx} = \frac{\sum y_i - b_{yx} \sum x_i}{n} = \frac{\sum x_i y_i - b_{yx} \sum x_i^2}{\sum x_i} \quad (13)$$

z levé strany lze určit, že mezi odhady a_{yx} a b_{yx} platí vztah

$$a_{yx} = \bar{y} - \bar{x} b_{yx} \quad (14)$$

Pro teoretické úvahy vyjadřujeme regresní koeficient značně jednodušeji, než výrazem (12). Dělíme-li čitatele i jmenovatele ve vzorci (12) druhou mocninou rozsahu souboru, získáme po úpravě vzorec

$$b_{yx} = \frac{\overline{xy} - \bar{x} \cdot \bar{y}}{\overline{x^2} - \bar{x}^2} \quad (15)$$

výraz ve jmenovateli je rozptyl nezávisle proměnné x ve výpočtovém tvaru

$$s_x^2 = \frac{\sum (x_i - \bar{x})^2}{n} = \overline{x^2} - (\bar{x})^2 \quad (16)$$

Výraz v čitateli je aritmetickým průměrem součinů jednotlivých hodnot obou proměnných zmenšený o součin aritmetických průměrů obou proměnných. Je to výpočtový tvar kovariance x na y , kterou značíme s_{yx} resp. s_{xy} . Kovarianci lze psát

$$s_{yx} = s_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n} = \overline{xy} - \bar{x} \cdot \bar{y} \quad (17)$$

Regresní koeficient b_{yx} lze tedy psát velmi jednoduše

$$b_{yx} = \frac{s_{xy}}{s_x^2} \quad (18)$$

Dosud uvedenné vzorce se používaly pro výpočet regresních koeficientů z neuspořádaných údajů. Tento výpočet je pro větší množství dat velmi pracný. Proto se používá výpočet z uspořádaných průměrů (podmíněných průměrů), nebo pomocí korelační tabulky. Tento systém výpočtu je o něco méně přesný než z neuspořádaných údajů, ale méně pracný.

6.4. Nelineární regrese dvou korelovaných veličin

V praxi se setkáváme také s takovými korelačními závislostmi kde není možné považovat jednu veličinu za nezávisle a druhou za závisle proměnnou. V tom případě se používají sdružené regresní přímky. Lineární závislost, s kterou jsme se zabývali dříve, přichází sice v praxi nejčastěji v úvahu, ale není jediným typem regrese v oblasti ekonomie a techniky, která se používá pro korelační zkoumání. Střetáváme se s případy, kdy nelze použít lineární zjednodušení, protože výsled-

ky by se nepřípustně zkreslily. Funkcí, pomocí kterých je možné v určitém intervalu vyjádřit některý z uvedených typů nelineární regrese, je zpravidla více. Při volbě regresní funkce musíme vycházet z těchto zásad:

- 1) Volba regresní funkce musí vždy vycházet z charakteru zkoumané korelační závislosti.
- 2) Volit takovou funkci, které konstanty lze lehce z empirických údajů vypočítat.
- 3) Při volbě musíme brát v zřetel rozsah souboru. Počet párů hodnot korelačních veličin musí být mnohonásobně větší než počet parametrů regresní funkce.

Funkce, které přicházejí z matematického hlediska do úvahy pro charakterizování formy nelineární závislosti jsou: parabola, hyperbola, exponenciála a logaritmická křivka. Při vyrovnání regresními křivkami používáme zase metodu nejmenších čtverců. rovnice paraboly n -tého stupně je

$$y' = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (19)$$

nebo rovnice hyperboly

$$y' = a_0 + a_1x^{-1} + a_2x^{-2} + \dots + a_nx^{-n} \quad (20)$$

Při výpočtu regresních koeficientů zase vycházíme ze stejných podmínek

$$\sum (y_i - y')^2 = F(a_0, a_1, a_2, \dots, a_n) = \min. \quad (21)$$

Funkci derivujeme parciálně podle všech proměnných (koeficientů a_i) a tyto derivace položíme rovny nule. Získáme tím soustavu rovnic pro tyto koeficienty. Jako příklad můžeme nejprve uvažovat kvadratickou regresi. V tomto případě dostaneme systém normálních rovnic

$$\sum (y_i - b_1 - b_2x_i - b_3x_i^2) \cdot 1 = 0 \quad (22)$$

$$\sum (y_i - b_1 - b_2x_i - b_3x_i^2) \cdot x_i = 0 \quad (23)$$

$$\Sigma(y_i - b_1 - b_2 x_i - b_3 x_i^2) * x_i^2 = 0 \quad (23)$$

který můžeme psát

$$\Sigma y_i - nb_1 - b_2 \Sigma x_i - b_3 \Sigma x_i^2 = 0 \quad (24)$$

$$\Sigma x_i y_i - b_1 \Sigma x_i - b_2 \Sigma x_i^2 - b_3 \Sigma x_i^3 = 0 \quad (25)$$

$$\Sigma x_i^2 y_i - b_1 \Sigma x_i^2 - b_2 \Sigma x_i^3 - b_3 \Sigma x_i^4 = 0 \quad (26)$$

Tento systém má jednoznačné řešení, jestliže alespoň 3 z hodnot x_i , $i=1,2,3,\dots,n$ jsou vzájemně různé. Z první rovnice získáme vztah pro odhad b_1 parametru

$$b_1 = \bar{y} - b_2 \bar{x} - b_3 \bar{x}^2 \quad (27)$$

Řešením rovnic pak dostaneme

$$b_2 = \frac{[n \Sigma x_i^4 - (\Sigma x_i^2)^2] * [n \Sigma x_i y_i - (\Sigma x_i)(\Sigma y_i)] - [n \Sigma x_i^3 - (\Sigma x_i^2)(\Sigma x_i)] * [n \Sigma x_i^2 y_i - (\Sigma x_i)(\Sigma y_i)]}{[n \Sigma x_i^2 - (\Sigma x_i)^2] * [n \Sigma x_i^4 - (\Sigma x_i^2)^2] - [n \Sigma x_i^3 - (\Sigma x_i)(\Sigma x_i^2)]^2} \quad (28)$$

$$b_3 = \frac{[n \Sigma x_i^2 - (\Sigma x_i)^2] * [n \Sigma x_i^2 y_i - (\Sigma x_i^2)(\Sigma y_i)] - [n \Sigma x_i^3 - (\Sigma x_i^2)(\Sigma x_i)] * [n \Sigma x_i y_i - (\Sigma x_i)(\Sigma y_i)]}{[n \Sigma x_i^2 - (\Sigma x_i)^2] * [n \Sigma x_i^4 - (\Sigma x_i^2)^2] - [n \Sigma x_i^3 - (\Sigma x_i)(\Sigma x_i^2)]^2} \quad (29)$$

z uvedených rovnic je zřejmé, že výpočet tímto způsobem je značně složitý zvláště pro polynomy vyššího řádu. Pro tyto výpočty se pak používá metody řešení soustavy rovnic pomocí maticového počtu. tato metoda je výhodná, používáme-li pouze polynomicickou regresi vyššího než 3. řádu. Potom se s výhodou používá výpočetní techniky.

6.5. Funkcionální regrese

Většina funkcí které používáme na regresní odhad patří do skupiny funkcí typu

$$y' = a_0 + a_1 F_1(x) + a_2 F_2(x) + \dots + a_n F_n(x) \quad (30)$$

kde $F_i(x)$ jsou funkce nezávisle proměnné x které neobsahují žádné další parametry. K tomuto typu patří : parabola, hyperbola, logaritmická funkce, ne však už exponenciála. Tyto funkce nazýváme lineárními funkcemi parametrů $a_0, a_1, \dots, a_n$. Při těchto funkcích můžeme na určitý regresní parametr použít metodu nejmenších čtverců. Pokud nelze použít funkce tohoto typu, tak používáme vhodné transformace, které převedou danou funkci na tento typ. Jedná se zde např. o tyto funkce :

a) Funkce typu $y' = ab^x$ (31)

která je exponenciální regresní funkcí. Použijeme zde transformace

$$z = \ln(y) \quad (32)$$

pak lze psát rovnici

$$\ln y' = \ln a + x * \ln b \quad (33)$$

a po dané substituci

$$z' = a' + x * b' \quad (34)$$

To už je známá lineární funkce a na ní lze použít známé vzorce pro parametry a, b ((12), (13)), ale pouze pro transformované veličiny x, z . Pak platí

$$a = ea' \quad (35)$$

$$b = eb' \quad (36)$$

b) Funkce typu $y' = a * e^{bx}$ (37)

pro kterou platí stejná transformace

$$z = \ln(y)$$
 (38)

Potom dojdeme k rovnici

$$\ln y' = \ln a + x * b$$
 (39)

a po substituci rovnici

$$z = a' + b * x$$
 (40)

dále platí stejné rovnice jako v bodě a) ((35),(36)). A potom platí

$$a = e^a$$
 (41)

c) Funkce typu $y' = a + b * \sqrt{x}$ (42)

kde je použita substituce

$$z = \sqrt{x}$$
 (43)

potom platí rovnice (12),(13), ale pro parametry z, y .

d) Funkce typu $y' = a + b / x$ (44)

kde je použita substituce

$$z = 1 / x$$
 (45)

a i zde platí rovnice (12),(13), ale pro parametry y, z

Je třeba si také uvědomit, že platí podmínka pro transformované, a ne pro původní proměnné

$$\Sigma(z - z')^2 = \min. \quad \text{a ne} \quad \Sigma(y - y')^2 = \min. \quad (46)$$

regresi založenou na nepřímém použití metody nejmenších čtverců, a vycházející z korelační závislosti původních proměnných k funkci těchto proměnných, nazýváme *funkcionální regrese*. Tento typ regrese se používá velmi často, a to pro svou jednoduchost, a v mnoha případech pro dostačující výstižnost.

6.6. Korelační úloha (měření těsnosti závislosti)

Druhou základní úlohou korelační analýzy je tzv. *korelační úloha*. Tato úloha spočívá v měření těsnosti (síly) závislosti zkoumaných veličin. Těsností rozumíme stupeň, s jakým se daná závislost blíží k funkční závislosti. Čím jsou empirické hodnoty méně rozptýlené okolo regresní čáry, tím menší je vliv vedlejších činitelů, a hovoříme o těsnější závislosti, jakmile jsou empirické hodnoty víc rozptýlené okolo regresní čáry, tak hovoříme o volnější závislosti.

Míry, které používáme na vyjádření stupně závislosti jsou:

- 1) Koeficient korelace- používá se u lineární závislosti
- 2) Index korelace- používá se při nelineární závislosti
- 3) Korelační poměr- používá se v případě, když neumíme vystihnout závislost určitou matematickou funkcí, resp. pokud bychom pro charakterizování potřebovali složité matematické funkce.

6.7. Měření těsnosti lineární závislosti

Jako míru těsnosti lineární závislosti používáme hodnotu, která se nazývá koeficient determinace. Je to poměr rozptylů odhadů závisle proměnné veličiny k celkovému rozptylu. Tuto veličinu značíme ryx^2 .

$$ryx^2 = \frac{s_y \cdot 2}{s_y^2}$$

(47)

kde je rozptyl odhadů s_y^2

$$s_y^2 = \frac{\sum (y_i - \bar{y})^2}{n} \quad (48)$$

a celkový rozptyl je součtem reziduálního rozptylu s_{yx} a rozptylu našich odhadů s_y^2

$$s_y^2 = s_y^2 + s_{yx}^2 \quad (49)$$

kde

$$s_{yx} = \frac{\sum (y_i - \bar{y})^2}{n} \quad (50)$$

koeficient determinace lze také stanovit jako součin obou regresních koeficientů

$$r_{yx} = b_{zx} * b_{xy} \quad (51)$$

Vprxi se ⁿma měření těsnosti lineární závislosti zpravidla nepoužívá koeficient determinace, ale jeho druhá odmocnina, kterou nazýváme koeficient korelace. Koeficient korelace se také někdy nazývá normovaný regresní koeficient. Udává, o kolik směrodatných odchylek se průměrně změní proměnná y , když se hodnota x zvětší o jednu směrodatnou odchylku.

$$r_{yx} = b_{yx} \frac{s_x}{s_y} \quad (52)$$

Výhoda normovaného regresního koeficientu oproti regresnímu koeficientu je v tom, že zatímco hodnota regresního koeficientu závisí na jednotkách měření, tak při normovaném

regresním koeficientu je tento vliv vyloučený.
Pro lineární regresi lze vyjádřit koeficient korelace v tvaru

$$r_{yx} = \frac{\overline{xy} - \bar{x} \cdot \bar{y}}{\sqrt{(\overline{x^2} - (\bar{x})^2)(\overline{y^2} - (\bar{y})^2)}} \quad (53)$$

nebo

$$r_{yx} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{[n \sum x_i^2 - (\sum x_i)^2][n \sum y_i^2 - (\sum y_i)^2]}} \quad (54)$$

Je možno říci, že čím více se korelační koeficient bude blížit k hodnotě 1 nebo -1, tím těsnější korelační vztah bude vyjadřovat, a naopak, čím více se bude blížit k hodnotě 0, tím volnější je volnější korelační vztah mezi zkoumanými veličinami.

6.8. Míra těsnosti jednoduché nelineární závislosti

Pro jednoduchou (párovou) nelineární regresi lze konstruovat míru těsnosti jako poměr rozptylu závisle proměnné k celkovému rozptylu závisle proměnné

$$\frac{s_{y'}^2}{s_y^2} = \frac{\bar{y}'^2 - (\bar{y})^2}{\bar{y}^2 - (\bar{y})^2} = I_{yx}^2 \quad (55)$$

Tuto míru jsme zde označili jako I_{yx}^2 a nazýváme ji indexem determinace, který bude při nezávislosti y na x roven 0, a při perfektní závislosti nabude hodnoty 1. K měření stupně závislosti jednoduché (párové) nelineární korelace se zpravidla používá jeho druhá odmocnina. Ta se pak nazývá *index korelace* a značí se I_{yx} . Z dosavadního výkladu vyplývá, že koeficient korelace r_{yx} je pouze zvláštním případem indexu korelace I_{yx} a to pro lineární korelaci y na x .

Volíme-li za regresní funkci nelineární funkci, je často možné, aby byla přetvořena na lineární regresní funkci pomocí vhodné transformace. Index korelace, a vztahy s ním související, se však neaplikují na původní proměnnou, ale na transformovanou proměnnou.

Při interpretaci indexu korelace je třeba pamatovat na to, že index korelace měří těsnost závisle na tom, jaká regresní funkce byla použita k vystižení dané závislosti. Případná nízká hodnota indexu korelace nemusí ještě ukazovat na to, že jde o malou intenzitu korelační závislosti, ale na to, že k vystižení povahy závislosti byla zvolena nevhodná regresní funkce.

V mé diplomové práci byly použity pro kvadratickou regresi tyto vztahy

$$r_{yx}^2 = 1 - \frac{s_{yx}^2}{s_y^2} \quad (56)$$

$$s_{yx}^2 = \frac{1}{n-1} \sum (y_i - \bar{y})^2 \quad (57)$$

$$s_y^2 = \frac{1}{n-1} \sum (y_i - \bar{y})^2 \quad (58)$$

$$\bar{y} = \frac{1}{n} \sum y_i \quad (59)$$

Toto jsou vztahy pro výpočet z velkých souborů, a proto se v nich počítá s hodnotou $n-1$.

7. Vyhodnocení naměřených veličin

Hodnoty naměřené při zkoušce motoru se zapisují do předtištěného formuláře " Záznam o zkoušce motoru ". Prvním úkolem při jejich vyhodnocení, je přepočítání všech hodnot na jednotky SI (zvláště tlaky) a dopočítání ostatních veličin. K tomu se používají tyto vzorce :

a) Přepočet tlaků d_{pT2} , d_{pD1} , d_{pCH} , d_p - zde je použit U manometr s náplní vody

$$d_p [\text{kPa}] = 0.00981 * d_p [\text{mm H}_2\text{O}] \quad (60)$$

b) Přepočet tlaků d_{pD2} , d_{pF} - zde je použit U manometr s náplní rtuti

$$d_p [\text{kPa}] = 10/75 * d_p [\text{mm Hg}] \quad (61)$$

c) Točivý moment M_t - konstanta dynamometru je zde $k=20/3$

$$M_t \text{ [Nm]} = k * M_t \text{ [údaj]} \quad (62)$$

d) Efektivní výkon P_e

$$P_e = \frac{M_t * n * \pi}{30\,000} \quad [\text{kW}] \quad (63)$$

kde se dosazuje M_t [Nm], n [1/min]

e) Střední efektivní tlak P_e

$$P_e = \frac{120 * P_e}{V_m * n} \quad [\text{MPa}] \quad (64)$$

kde se dosazuje P_e [kW], V_m [dm³], n [1/min]

f) Tlaky za dmychadlem

$$P_{D2kin} = 9.7 * \frac{m_v^2 * T_{D2}}{P_{D2}} \quad (65)$$

kde se dosazuje m_v [kg/s], P_{D2} [kPa]

$$P_{D2celk} = dP_{D2} + P_{D2kin} \quad (66)$$

g) Měrné spotřeby paliva

$$m_{pe} = \frac{M * 3600}{P_e * \tau} \quad [\text{g/kWh}] \quad (67)$$

$$m_{PK} = \frac{M * 120}{n * i * \tau} \quad [g/cy] \quad (68)$$

$$m_{PK}' = \frac{M * 120}{n * i * \tau * V_m} \quad [g/cy.l] \quad (69)$$

kde M [g] , τ [s] , P_e [kW] , n [1/min] , V_m [dm³]

e) Hustota vzduchu

$$\rho_{vo} = \frac{p_o}{287.1 * (t_o + 273.15)} \quad [kg/m^3] \quad (70)$$

$$\rho_{vPL} = \frac{(d_{pp} + p_B)}{287.1 * (t_{p3} + 273.15)} \quad [kg/m^3] \quad (71)$$

kde p_o [kPa] , d_{pp} [kPa] , p_B [kPa] , t_o [°C] , t_{p3} [°C]

f) Spotřeba vzduchu

$$M_v = 3.51026 * 10^{-3} * \left(1 - \frac{dp}{1.8 p_1}\right) * \sqrt{dp * \rho_{PL}} \quad (72)$$

$$m_{vx} = \frac{M_v * 120}{n * i} \quad [\text{kg/cy}] \quad (73)$$

$$m_{vx}' = \frac{M_v * 120}{n * i * V_m} \quad [\text{kg/cy.l}] \quad (74)$$

kde d_p [kPa] , p_1 [kPa] , v_{pL} [kg/m³] , n [l/min]
 V_m [dm³]

g) Přebytek vzduchu

$$M_p = \frac{M}{1000 * \tau} \quad [-] \quad (75)$$

$$\lambda = \frac{M_v}{14.32 * M_p} \quad [-] \quad (76)$$

kde M [g] , τ [s] , M_v [kg/s]

h) Stlačení dmychadla π

$$\pi = \frac{d_{pD2}}{d_{pD1}} \quad [-] \quad (77)$$

kde d_{pD1} [kPa] , d_{pD2} [kPa]

Po ukončení výpočtů je nutná jejich kontrola a případné přepočítání. Je-li ve výsledné hodnotě chyba, a nevznikla-li ve výpočtu, je nutné se podívat na naměřené veličiny, protože chyby u zkoušek motoru často vznikají již při chybném měření hodnot, nebo při jejich zápisu. Po úplném doplnění *Zkušebního protokolu* se zpravidla přistupuje k vynesení naměřených a vypočtených hodnot do grafů – sestrojení charakteristik. Používají se zde všechny charakteristiky popsané v kapitole 5. této diplomové práce. Při použití výpočetní techniky se dále určují regresní rovnice, případně se vykreślují grafy. Moje diplomová práce má za účel shromáždit všechny tyto etapy do jednoho programu tak, aby uživatel pouze zadal naměřené hodnoty, zvolil výstupní veličiny, a výstupem byl vytištěný protokol nebo zvolené charakteristiky motoru. Tím by se měla práce při vyhodnocování zkoušek motoru značně zkrátit a zjednodušit.

8. Programy používané pro vyhodnocení výsledků

Na vyhodnocování naměřených výsledků se zpravidla používají různé speciální programy. Pro můj problém přichází v úvahu programy, které jsou schopné rozsáhlejších výpočtů, a které mají zároveň možnost zobrazení výsledných hodnot ve formě grafu. Tyto programy jsou však velmi složité, a jsou orientovány spíše univerzálně. K nejznámějším a nejrozšířenějším programům u nás patří **Lotus 1-2-3** a **Quattro**. Tyto programy se označují jako tabulkové procesory nebo spreadsheet. Tyto programy slouží k plánování, ekonomickým výpočtům, vytváření tabulek a k výpočtům, které se dříve prováděly pomocí kalkulaček. Tato kategorie programů není určena k programování ale hlavně k využití daného editoru. Lze v nich však vytvářet určité programy (spíše makra). Základem každého spreadsheetu je čtvercová síť. Tato síť se nazývá spreadsheet a bývá značně rozsáhlá. Tato síť se skládá z buněk, které jsou jednoduchým způsobem adresované. Do každé z

buňek se vkládají data, a to buď z klávesnice nebo jsou výsledkem určitých matematických nebo statistických operací s ostatními buňkami. Výstupní údaje se zobrazují buď ve formě tabulky, nebo je možné určitá data zobrazit ve formě grafu. Tyto programy mají možnost zobrazení výstupních hodnot do grafů různých typů.

Program **Lotus 1-2-3** se stal světovým standartem. Má možnost zobrazení výstupních veličin v grafech těchto typů : čárový graf, 2 typy histogramu, kruhový graf a graf funkční závislosti XY. V grafu XY lze zobrazit až 5 funkčních závislostí. Tyto grafy lze vytisknout. Tento program však umí zakreslené body charakteristiky pouze spojit přímkami, neumí použít matematickou regresii.

Program **Quattro** od firmy Borland international patří také do této kategorie - je to tabulkový procesor (spreadsheet). Z hlediska kvality, výkonu a uživatelského komfortu se jeví jako optimální, a překonává standart stanovený Lotusem 1-2-3. Oproti tomuto programu má vyšší rychlost a dokonalejší grafiku, ale i další zlepšení možností využití. Kombinuje 3 typy programů a to spreadsheet, databázi a grafiku. Umožňuje tím velké možnosti pro práci s tímto programem. Můžeme zde zobrazit data až v 10 typech grafu - čárový, histogram, 3-Dimenzionální, koláčový, Tento program umožňuje vytváření tabulek a grafů vysoké kvality. Je schopen zobrazovat a tisknout i grafy funkční závislosti XY, ale zobrazené body opět pouze spojí přímkami.

Další programy této kategorie jsou Excel, Supercalc 5.0, Lucid 3-D, Surpass, atd. Tyto programy mají vysokou kvalitu, ale tím i značné ceny za jejich nákup. Dalším problémem je naučit pracovat s těmito programy všechny jeho uživatele. To si většinou vyžaduje, aby tito pracovníci prošli školením u specializované firmy. Proto se často jeví jako lepší možnost vytvoření speciálního programu, který by pro daný účel splňoval všechny požadavky (někdy jsou tyto programy i rychlejší a lépe ^lspňují naše požadavky).

V podnicích, které se zabývají výzkumem a vývojem spalova-

cích motorů se používají různé způsoby vyhodnocování těchto motorů při zkouškách. Jednou cestou je vybavení zkušebního stanoviště dynamometry řízenými počítači řady PC - to umožňuje nejen kvalitní řízení vlastní zkoušky, ale i rychlé vyhodnocení výsledků a jejich zpracování. Vyžaduje to však speciální technické a programové vybavení, které je velmi nákladné, a může si ho dovolit pouze málo podniků nebo výzkumných ústavů. Starší typy zkušebních zařízení využívá sice řízení dynamometru a sběru dat počítačem, ale naměřená data vydává pouze ve formě protokolu o zkoušce, a jsou nahrána na některý druh paměti. Většinou nemají možnost grafického výstupu ve formě charakteristik.

Tento druhý způsob využívají v VVÚ Škoda Mladá Boleslav. Mají zde zkušební zařízení firmy Hoffmann, které má výstup pouze ve formě protokolu. Pro kreslení charakteristik používají vlastní program, který využívá naměřená data. Tato data však musí přenášet z páskové paměti na diskety na jiném počítači. Na zpracování grafů používají pak vlastní program nebo některý z tabulkových procesorů.

9. Popis vytvořeného programu

Při tvorbě programu k mé diplomové práci jsem použil programovacího jazyka Turbo Pascal. Při výběru programovacího jazyka jsem Basic zavrhl brzy, pro jeho známé velké nedostatky - je pomalý a při velkém rozsahu programu je značně nepřehledný. Turbo Pascal má oproti tomu značné výhody - jeho rychlost je dostatečná, je to strukturovaný jazyk, a tím lze dosáhnout velké přehlednosti programu a snadnějšího odlaďování. Jiné programovací jazyky jsem nemohl použít, protože podmínkou mé diplomové práce bylo použití výpočetní techniky na katedře KSD a zde není jiný programovací jazyk zakoupen. K programování jsem použil editor TurboPascal ver. 5.5 od firmy Borland international. Při své práci

jsem se snažil maximálně využít výhod tohoto programovacího jazyka - vytváření procedur a použití strukturovaných příkazů.

Popis použitých procedur

InKonst - procedura pro zadávání konstant programu - konstant měřících přístrojů (tlakoměrů, clonky a brzdy).

InDataKey - procedura pro zadávání vstupních hodnot z klávesnice. Data se zapisují do textového souboru S01.

InDataDisc - procedura pro zadávání vstupních dat čtením z disku nebo diskety. Data se čtou ze souboru S02 a zapisují do souboru S01.

CLS - procedura mazání celého stínítka obrazovky.

OprDat - procedura na opravu vstupních dat. Tato procedura čte data ze souboru S01 a ukládá si data do souboru S04. Po opravě celého souboru se data ze S04 přepíší do S01. Tato procedura potřebuje pro svou práci vytvořit na disku adresář a do příkazu Assign() zadat jeho jméno a path.

Vypocet - tato procedura slouží k výpočtu výstupních hodnot. Čte data ze souboru S01 a výsledky ukládá do souboru S03, který musí být na disku opět vytvořen a musí být zadána cesta (path) v příkazu Assign().

Protokol - slouží k organizování výpisu protokolu. Přepisuje postupně data ze souboru S03 do proměnné VY [i,j] a potom volá proceduru Vypis.

Vypis - tato procedura provádí výpis protokolu o zkoušce motoru. Využívá proměnnou VY [i,j] a protokol vypisuje na obrazovku v režimu GRAPH. Tisk protokolu se provádí po jednotlivých stránkách externí procedurou Tisk.pas. Tato procedura je připojena k programu pomocí direktivy {\$I Tisk.pas}.

ZOsy - tato procedura slouží pro tisk grafu - zadávají se zde veličiny, s kterými bude provedena regrese, a které budou vytištěny. Využívá soubor S03, z kterého čte data, a zapisuje je do proměnných X, Y1, případně Y2 (podle typu grafu).

LReg1 - procedura pro regresi (31)

tento program využívá pouze standartní unity TurboPascalu - GRAPH, CRT, SYSTEM, PRINTER. Jako externí využívá pouze proceduru TISK.pas. Při výpočtu však využívá matematický koprocessor 80x87 nebo si provádí jeho emulaci programovými prostředky. Toto je zajištěno direktivou {\$E+} a {\$N+}. Program je při své práci dostatečně rychlý (na počítači PC-AT není výpočet rozpoznatelný zdržením) a rychlý je i grafický výstup. Jediné zdržení je při provedení tisku obrazovky na znakové tiskárně. Je to způsobeno strukturou procedury TISK, která však ve standartním TurboPascalu není, a musela být zvlášť vytvořena. Další možností bylo použití speciálních procedur, které pod názvem GRAPHIX dodává firma Borland. Tyto procedury jsou však zastaralé a neslučitelné s jednotkou GRAPH. Navíc pracují pouze s grafickou kartou EGA a mají i jiné nedostatky. Použití programu PIZZAS nebylo možné z důvodu nedostatku paměti. Proto jsem využil pomalejší vlastní proceduru.

10. Závěr

Výpočtový program je vytvořen pro motor M 2.4 na měřicím stanovišti, používaném na KSD. Pro jiné měřicí přístroje stačí zadat jiné konstanty do programu. Při použití jiné měřicí metody by se musela změnit ta část programu sloužící danému výpočtu (jedná se např. o jiné měření spotřeby,...). Program je však vytvořen tak, aby byl schopen zpracovat i výsledky měření na jiných motorech. Stačí, aby se místo hodnot, které nebyly měřeny zadala nula. Program potom přeskočí tu část výpočtu, kde by výpočet s nulovou hodnotou způsobil chybu při běhu programu a za výsledek dosadí nulu. Ošetřeno je také zadávání hodnot pro regresní analýzu, kde tyto nesmí nabývat hodnot nula a menších než nula z toho důvodu, protože používám pro regresi logaritmickou funkci. Výpočet hodnot jiného motoru než M 2.4 je jako příloha této diplomové práce, kde se jedná o motor Zetor Z 7701. Byly použity podklady ze zkoušek tohoto motoru při Vyzkumném úkolu číslo 60 054/370, kde byl zjišťován vliv laserové úpravy na vložku válce. Jako příklad jsem použil naměřené otáčkové a zatěžovací charakteristiky.

Pokud se program používá pro jiné motory, je nutné přizpůsobit se označení vstupních a výstupních veličin. Např. u motoru Z 7701 byla měřena teplota výfukových plynů a ta je v protokolu označena jako t_{r2} (teplota za turbínou) a teplota sání, která je zde označena t_p (teplota plnicí).

V přílohách k mé diplomové práci jsou příklady Zkušebních protokolů a příslušných charakteristik. Provedení Záznamu o zkoušce motoru je dobré, provedení charakteristik je také dobré, ale s chybami tisku. Tyto jsou způsobeny chybou při komunikaci počítače a znakové tiskárny, a nelze je zatím odstranit. Velikost grafu je maximální možná tímto způsobem tisku a jeho kvalita je omezena rozlišovací schopností grafické karty a použité tiskárny.

Je také zřejmé, že použité typy regresních funkcí vyhovují pro většinu možných závislostí. Vyjímkou je asi průběh měrné

spotřeby, pro který by bylo nutné použít polynomicickou regresi alespoň 5.řádu a nebo provést rozdělení křivky v místě minimální hodnoty na dvě křivky.

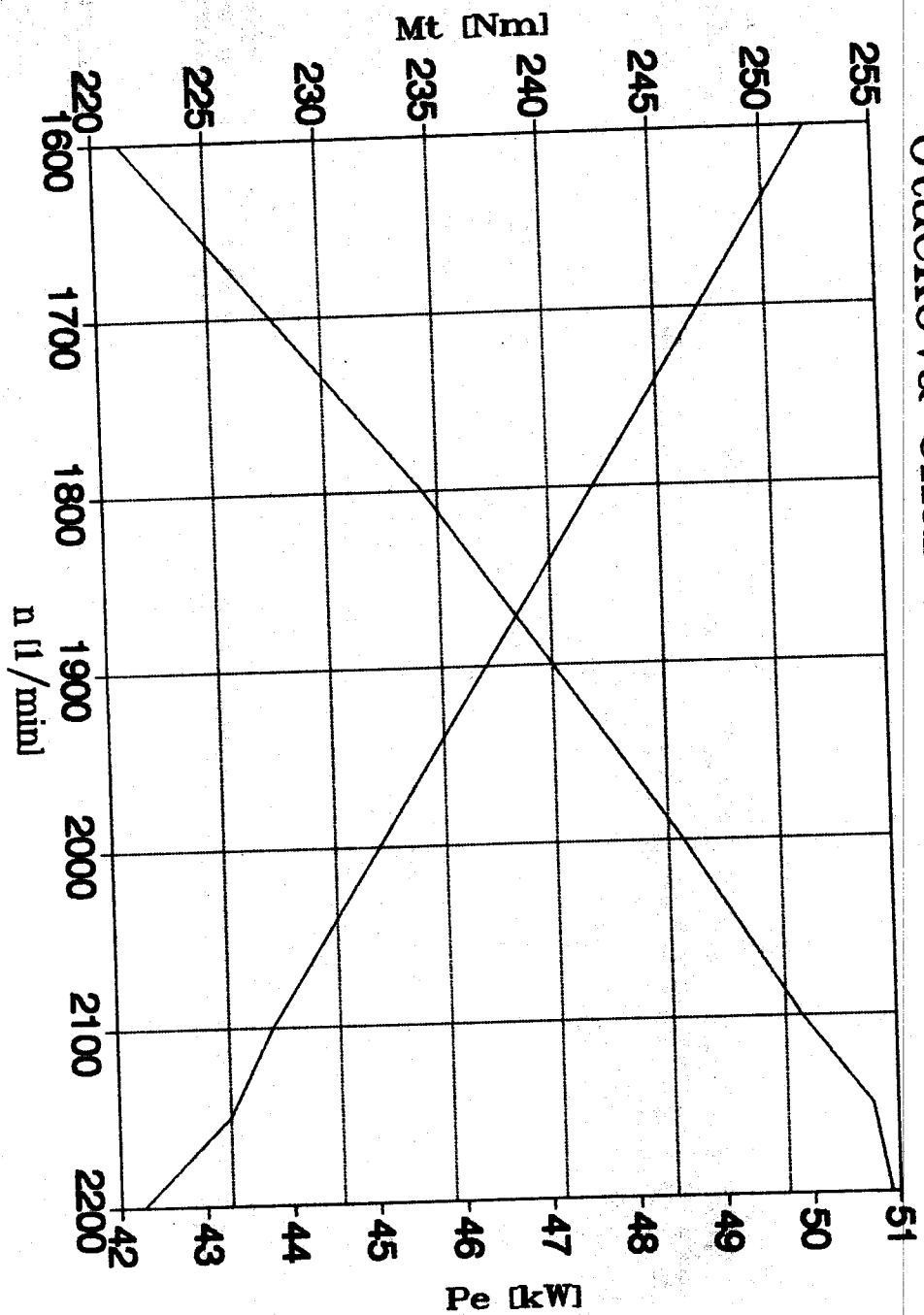
Pro porovnání kvality grafů jsem k diplomové práci přiložil i graf vytvořený pomocí programu Quattro. Pro vyhodnocování výsledků na KSD bude však asi většinou stačit mnou vytvořený program, protože jeho obsluha je nenáročná a lze ji snadno zvládnout. Všechny pokyny k obsluze programu jsou v tomto programu zabudovány.

Mnou vytvořený program podle mne splňuje požadavky zadané v mé diplomové práci. Jako příloha k ní je výpis programu i použité procedury TISK, příklady charakteristik i se Záznamy o zkoušce motoru a příklad grafu vytvořený programem QUATTRO.

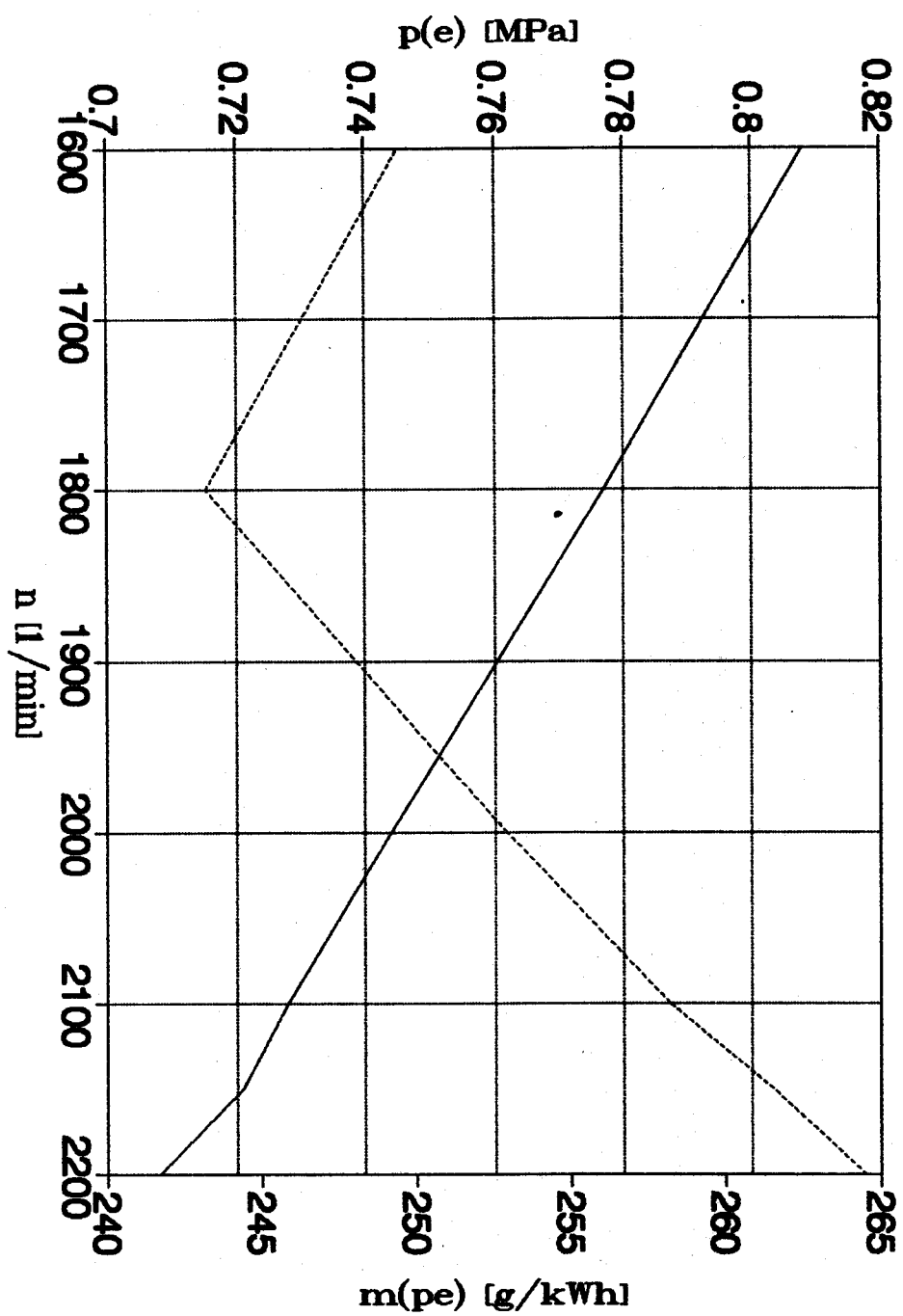
11. Použitá literatura

- 1\ Linczeni, A. - Inženýrská statistika. Alfa, Bratislava 1973
- 2\ Cyhelský, L. - Úvod do teorie statistiky. SNTL, 2. vyd.,
Praha 1981
- 3\ Kožoušek, J. - Teorie spalovacích motorů. SNTL, 2. vyd.,
Praha 1971
- 4\ Burgr, M. - Lotus 1-2-3, uživatelská příručka. TIS JZD Horní
Branná
- 5\ Manual QUATTRO
- 6\ Výzkumná zpráva SM 249/90. VŠST Liberec, 1990
- 7\ Záznamy o zkoušce motoru -030, 031/1, 032/2, 033. KSD VŠST
Liberec

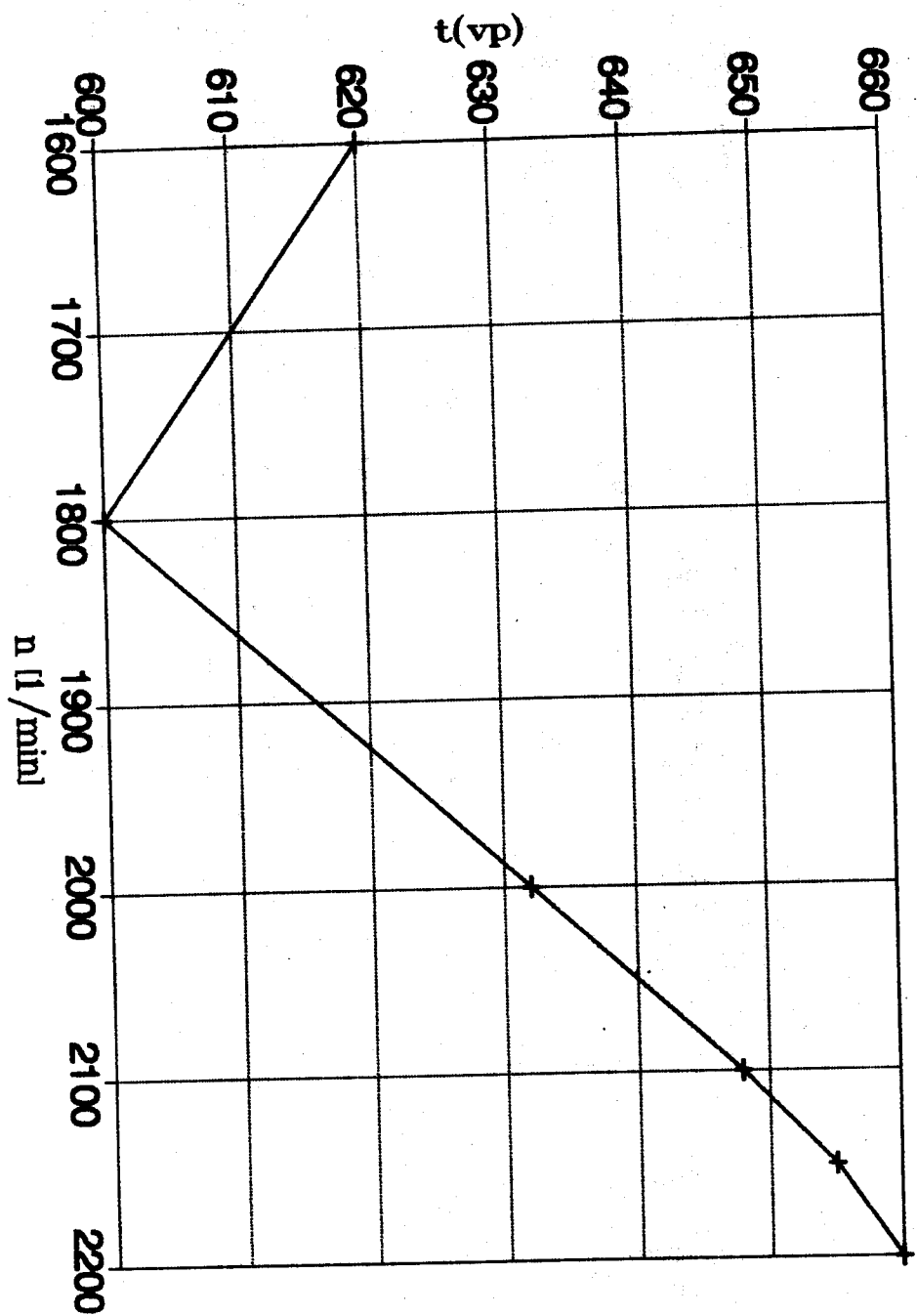
Otackova charakteristika Z 7701



Otackova charakteristika Z 7701



Otackova charakteristika Z 7701



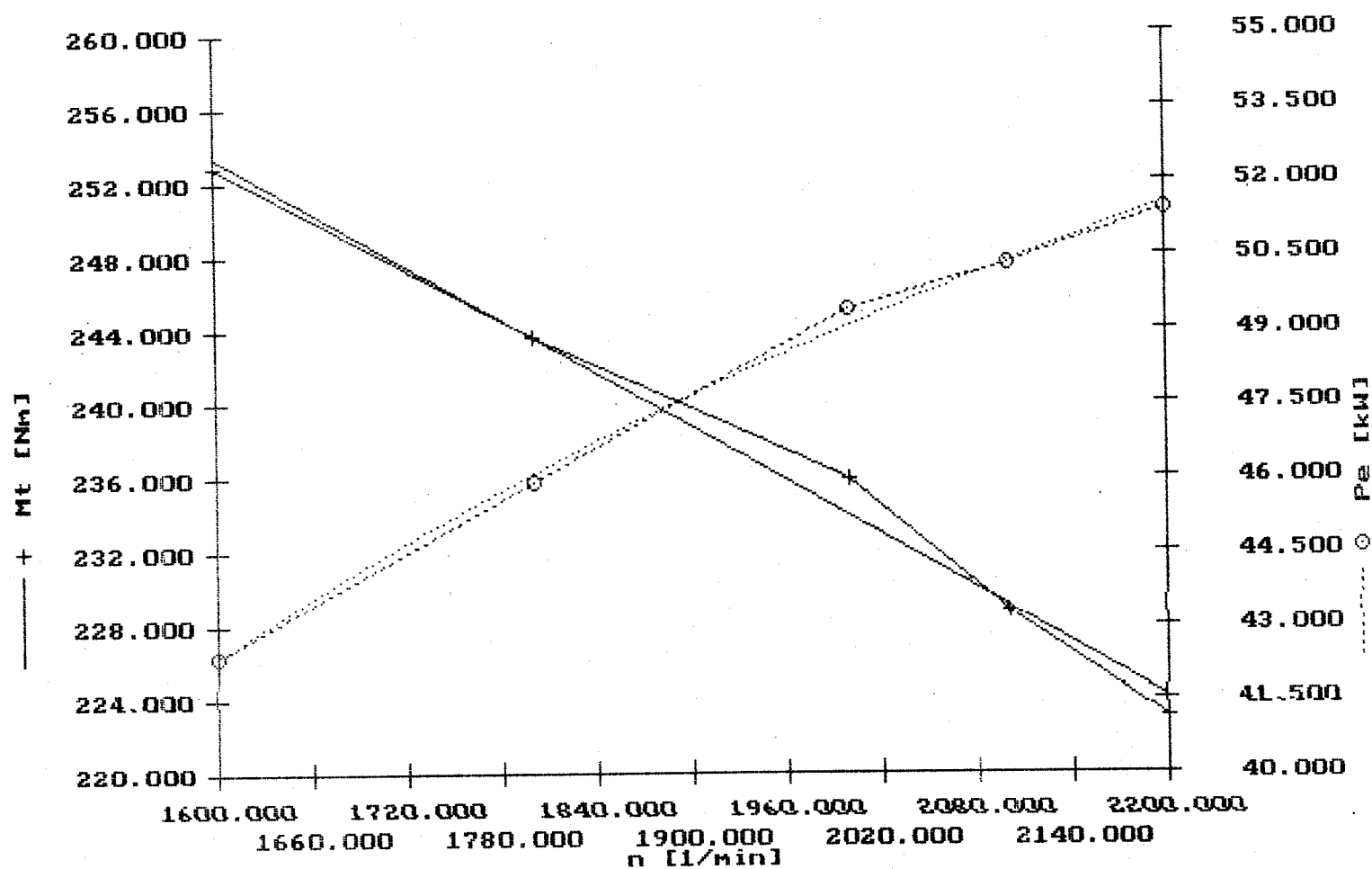
ZAZNAM O ZKOUSCE MOTORU

Typ motoru:	Z 7701	Jmenovity výkon:	50	
Jmenovité otacky:	2200	Ustrik. čerpadlo:		
Trysky:		Turbodmychadlo:		
Uyr.cislo motoru:		Predvstrik:		
Zdvihovy objem:	3.922	Pocet valcu:	4	
Barometricky tlak:	97.267	Teplota vzduchu:	20.5	
Vlhkost vzduchu:		Datum:	17.8.1990	
Cislo mereni	1	2	3	4
n [l/min]	1600.000	1800.000	2000.000	2100.000
n(t) [l/min]	0.000	0.000	0.000	0.000
Mt [Nm]	252.810	243.580	235.888	228.709
t(o)	93.000	93.000	93.000	94.000
t(w)	83.000	83.000	83.000	83.000

dp(p) [kPa]	0.000	0.000	0.000	0.000
dp(ch) [kPa]	0.000	0.000	0.000	0.000
dp [kPa]	0.000	0.000	0.000	0.000
k [%]	0.000	0.000	0.000	0.000
Pe [kW]	42.359	45.914	49.404	50.296
p(e) [MPa]	0.810	0.780	0.756	0.733
n(pe) [g/kWh]	248.649	243.276	248.358	256.639
n(px) [g/cy]	0.055	0.052	0.051	0.051
n(px) [g/cy.l]	0.014	0.013	0.013	0.013
Ro(v0) [kg/m^3]	1.154	1.154	1.154	1.154
Ro(vp1) [kg/m^3]	0.000	0.000	0.000	0.000
n(vx) [kg/cy]	0.000	0.000	0.000	0.000
n(vx) [kg/cy.l]	0.000	0.000	0.000	0.000
Lambda	0.000	0.000	0.000	0.000
Stlaceni Pi	1.000	1.000	1.000	1.000
dp(f) [kPa]	0.000	0.000	0.000	0.000

t(1T1)	0.000			
t(2T1)	0.000			
t(T2)	660.000			
t(D1)	25.000			
t(D2)	0.000			
t(D3)	0.000			
t(D3)	0.000			
t(p)	0.000			
t(w1)	0.000			
t(w2)	0.000			
p(o) [MPa]	0.000			
dp(T1) [kPa]	0.000			
dp(T2) [kPa]	0.000			
dp(D1) [kPa]	0.000			
dp(D2) [kPa]	0.000			
p(D2kin.) [kPa]	0.000			
p(D2celk.) [kPa]	97.267			

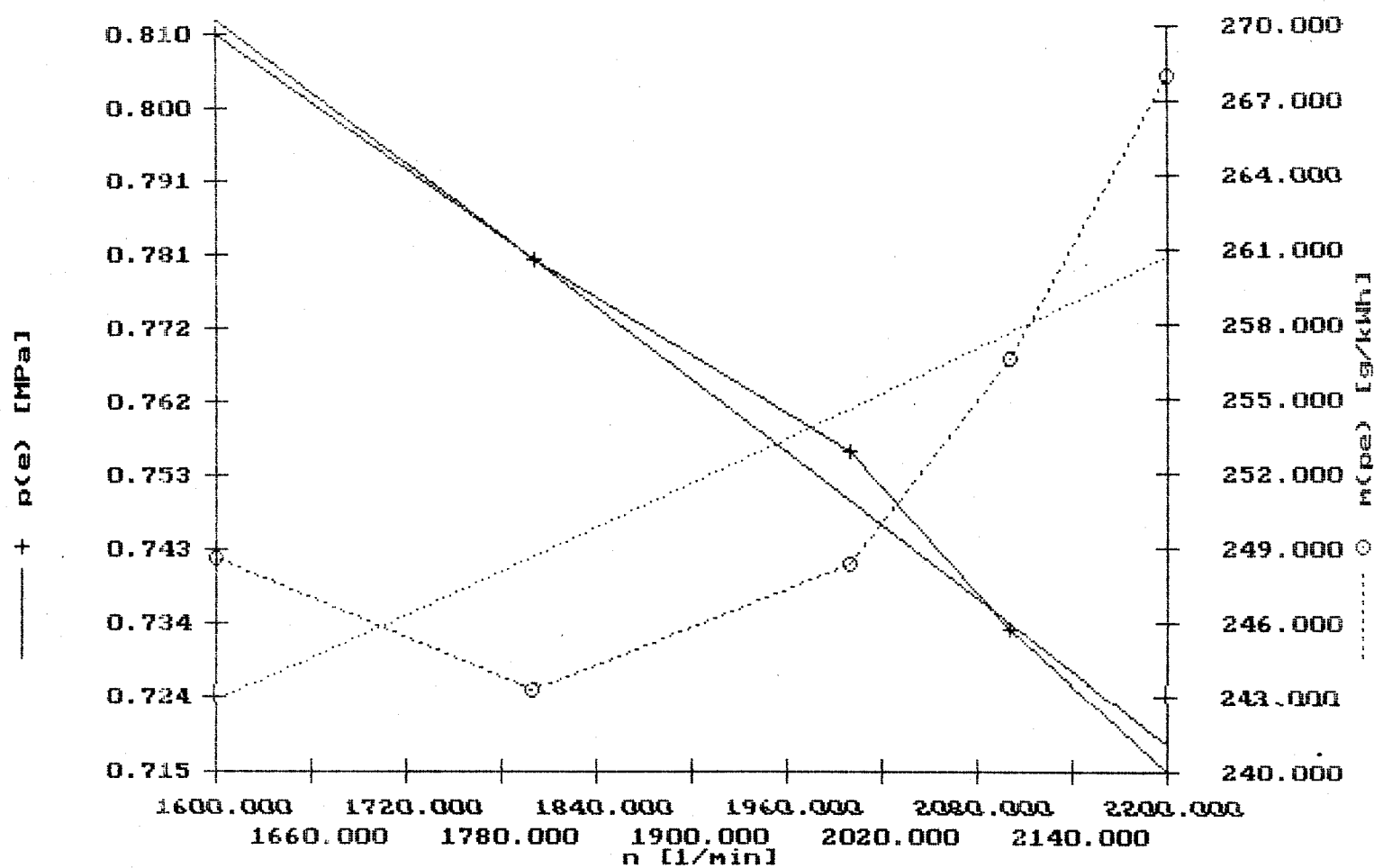
Otackova charakteristika Z 7701



$$\begin{aligned}
 \text{---} & \quad Mt [Nm] = 331.406 + -0.049 * n [1/min] \\
 \text{---} & \quad Pe [kW] = 75.988 + -53833.676 / n [1/min]
 \end{aligned}$$

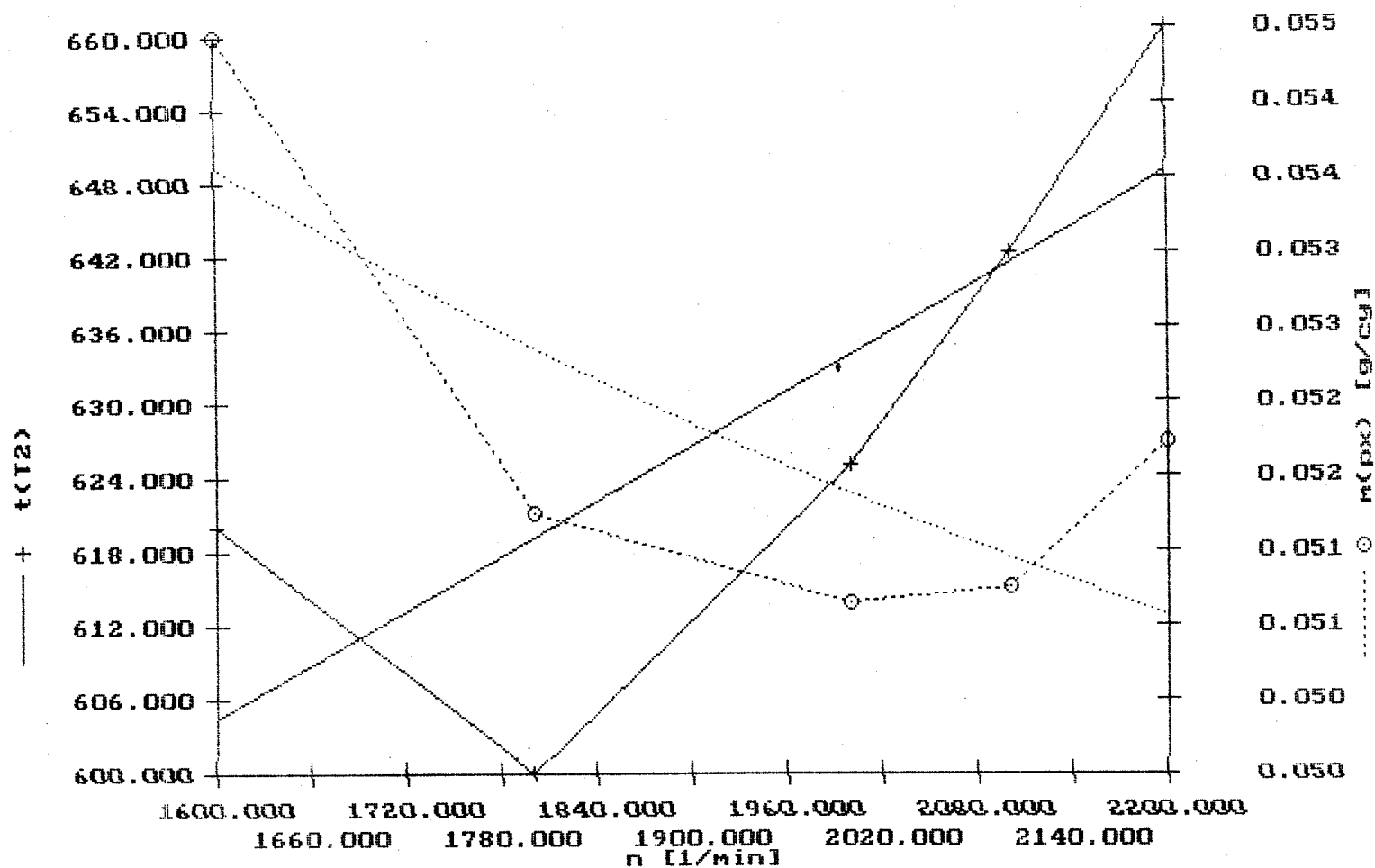
$$\begin{aligned}
 r &= -0.995 \\
 r &= -0.999
 \end{aligned}$$

Otackova charakteristika Z 7701



$$\begin{aligned}
 p(e) \text{ [MPa]} &= 1.062 + -0.000 * n \text{ [1/min]} & r &= -0.995 \\
 n(pe) \text{ [g/kWh]} &= 200.865 * 1.000 ^ n \text{ [1/min]} & r &= 0.757
 \end{aligned}$$

Otackova charakteristika Z 7701



$t(T2) = 487.078 + 0.073 * n$ [1/min]
 $\eta(px) = 0.043 + 17.382 / n$ [1/min]

$r = 0.777$
 $r = 0.778$

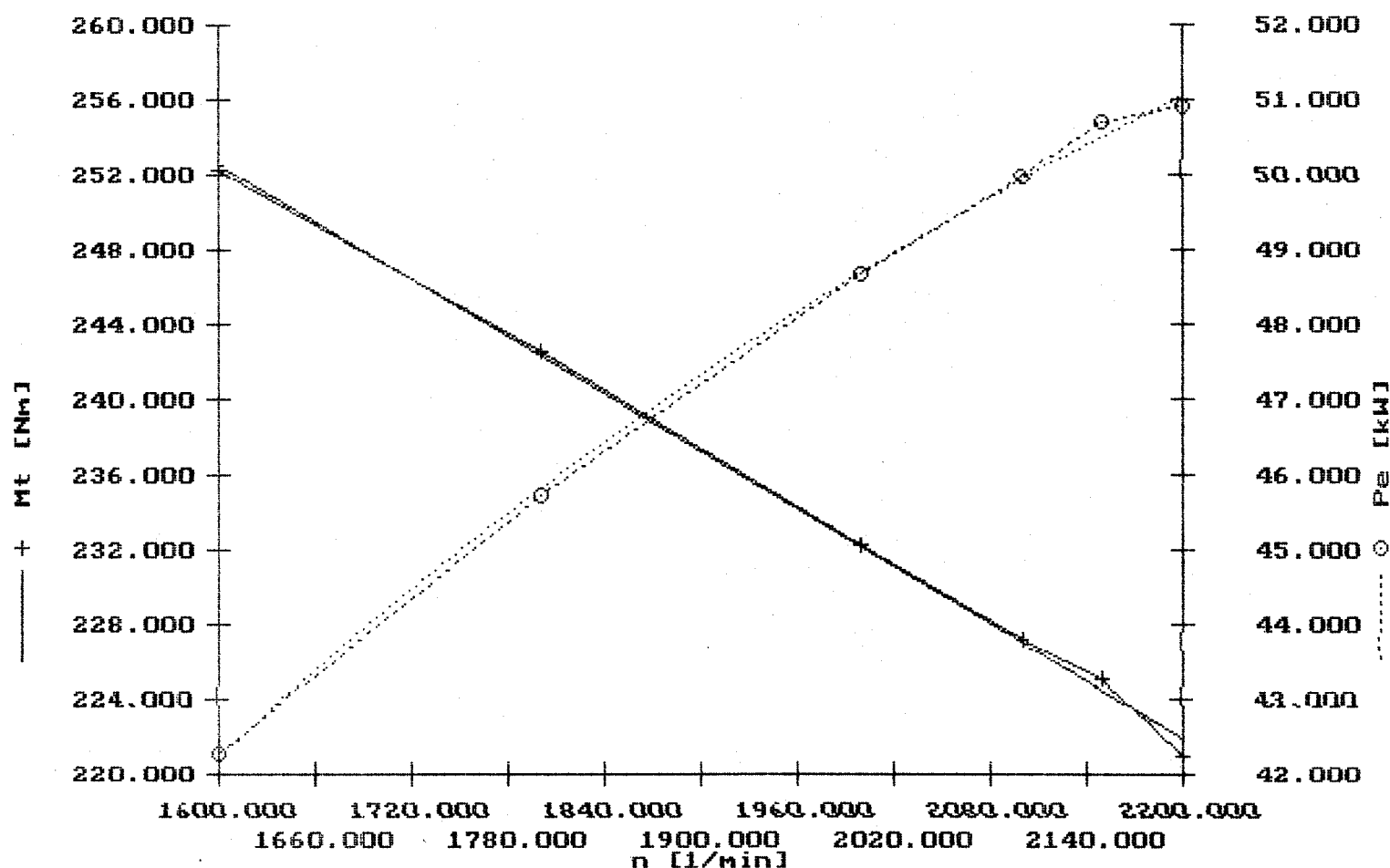
ZAZNAM O ZKOUSCE MOTORU

Typ motoru:	Z 7701	Jmenovity vykon:	50	
Jmenovite otacky:	2200	Ustrik. cerpadlo:		
Trysky:		Turbodmychadlo:		
Uyr.cislo motoru:		Predvstrik:		
Zdvihovy objem:	3.922	Pocet valcu:	4	
Barometricky tlak:	97.400	Teplota vzduchu:	22	
Vlhkost vzduchu:		Datum:	7.8.1990	
Cislo mereni	1	2	3	4
n [1/min]	2200.000	2150.000	2100.000	2000.000
n(t) [1/min]	0.000	0.000	0.000	0.000
Mt [Nm]	221.017	225.119	227.170	232.298
t(o)	90.000	91.000	91.000	91.000
t(w)	82.000	82.000	82.000	82.000

dp(p) [kPa]	0.000	0.000	0.000	0.000
dp(ch) [kPa]	0.000	0.000	0.000	0.000
dp [kPa]	0.000	0.000	0.000	0.000
k [%]	0.000	0.000	0.000	0.000
Pe [kW]	50.919	50.685	49.957	48.652
p(e) [MPa]	0.708	0.721	0.728	0.744
m(pe) [g/kWh]	264.501	261.513	258.192	252.972
m(px) [g/cy]	0.051	0.051	0.051	0.051
m(px) [g/cy.l]	0.013	0.013	0.013	0.013
Ro(v0) [kg/m^3]	1.149	1.149	1.149	1.149
Ro(vp1) [kg/m^3]	1.130	1.130	1.130	1.130
m(vx) [kg/cy]	0.000	0.000	0.000	0.000
m(vx) [kg/cy.l]	0.000	0.000	0.000	0.000
Lambda	0.000	0.000	0.000	0.000
Stlaceni Pi	1.000	1.000	1.000	1.000
dp(f) [kPa]	0.000	0.000	0.000	0.000

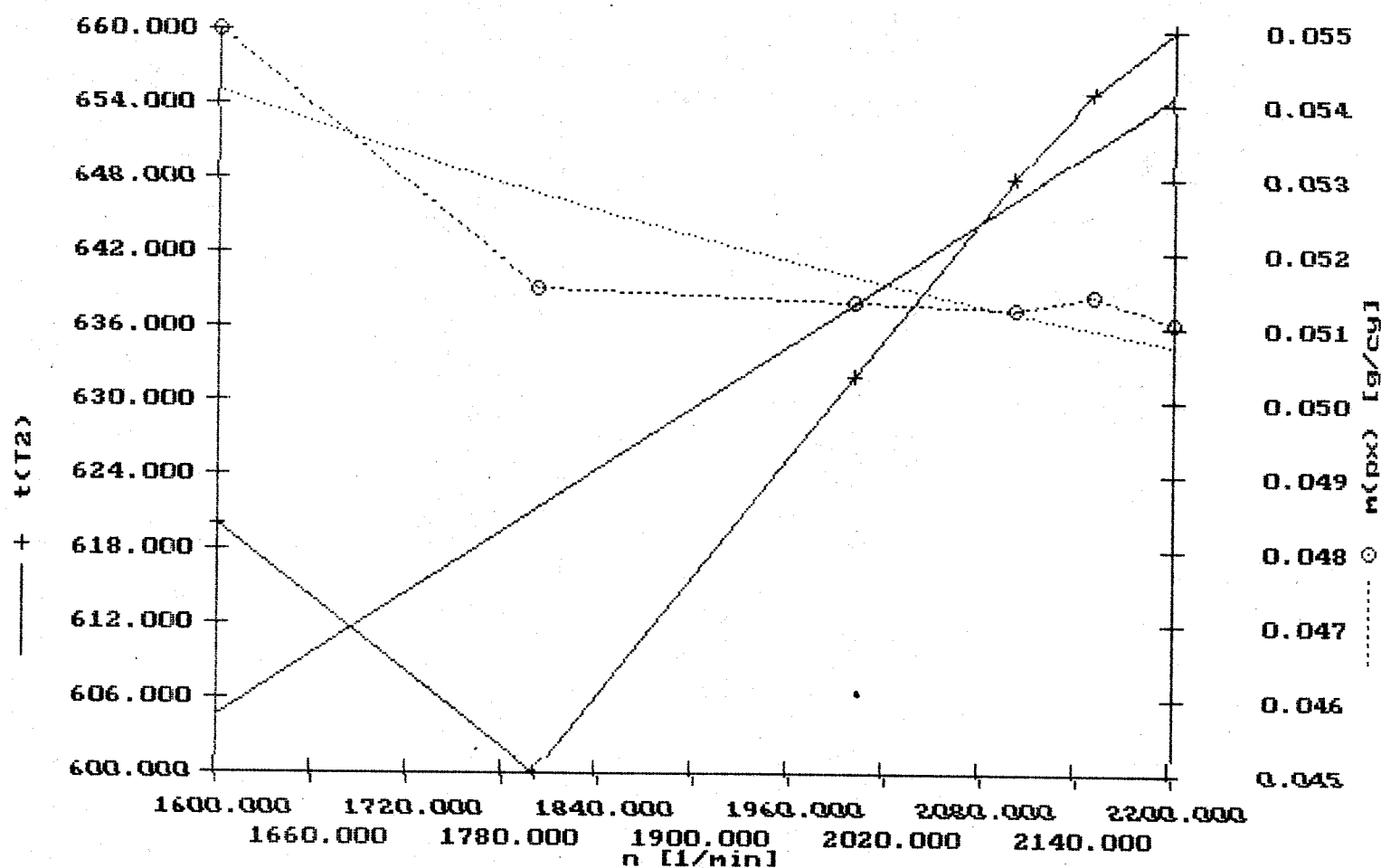
t(1T1)	0.000	0.000		
t(2T1)	0.000	0.000		
t(T2)	600.000	620.000		
t(D1)	0.000	0.000		
t(D2)	0.000	0.000		
t(D3)	0.000	0.000		
t(p)	27.000	27.000		
t(w1)	0.000	0.000		
t(w2)	0.000	0.000		
p(o) [MPa]	0.000	0.000		
dp(T1) [kPa]	0.000	0.000		
dp(T2) [kPa]	0.000	0.000		
dp(D1) [kPa]	0.000	0.000		
dp(D2) [kPa]	0.000	0.000		
p(D2kin.) [kPa]	0.000	0.000		
p(D2celk.) [kPa]	97.400	97.400		

Otackova charakteristika Z 7701



$$\begin{aligned}
 \text{---+---} \quad Mt \text{ [Nm]} &= 334.311 + -0.051 * n \text{ [1/min]} & r &= -0.999 \\
 \text{...o...} \quad Pe \text{ [kW]} &= 74.590 + -51791.977 / n \text{ [1/min]} & r &= -0.999
 \end{aligned}$$

Otackova charakteristika Z 7701



procedure error(chyba:integer);

begin
end;

procedure tisk;

var Bajt:byte;

Lst:text;

chyba:integer;

ymax,xmax,y1,x1:word;

amod,db,hb,pocrad,rad:byte;

begin

xmax:=getmaxx;

ymax:=getmaxy;

Assign(lst,'PRN');

Rewrite(lst);

Chyba:=IOResult;

if Chyba<>0 then

begin

Error(Chyba);

Exit

end;

Case XMax of

319 : amod:=5;

639 : amod:=4;

719 : amod:=2;

end;

pocrad:=(YMax div 8)-1;

hb:=XMax div 256;

db:=XMax mod 256;

write(lst,#27,#51,#23);

Chyba:=IOResult;

if Chyba<>0 then

begin

Error(Chyba);

Exit

end;

for rad:=0 to pocrad do

begin

write(lst,#27,#42,char(amod),char(db),char(hb));

Chyba:=IOResult;

if Chyba<>0 then

begin

Error(Chyba);

Exit

end;

for x1:=0 to XMax-1 do

begin

bajt:=0;

for y1:=0 to 7 do

if GetPixel(x1,rad*8+y1)<>Black

then

bajt:=bajt*2+1

else

bajt:=bajt*2;

write(lst,char(bajt));

Chyba:=IOResult;

if Chyba<>0 then

begin

Error(Chyba);

Exit

end;

end;

for j:=1 to 20 do

write(lst,#0);

write(lst,#10);

end;

write(lst,#27,#50,#10);

Chyba:=IOResult;

```
begin
  Error(Chyba);
  Exit
end;
Close(1st);
Chyba:=IOResult;
if Chyba<>0 then
begin
  Error(Chyba);
  Exit
end;
{!+}
end;
```

Program MOTOR;

```
(#####  
#####  
#####  
##### ADAMEC Martin , VSST LIBEREC , KSD ,1991 #####  
#####  
#####  
#####)
```

{%N+} {%IFDEF CPU97} {%E+} {%ENDIF}

```
uses Crt,Printer,Graph;  
label 1,3;  
type soubor=text;  
var so1,so2,so3,so4:soubor;  
  ad:array[1..27] of real;  
  as:array[1..27] of string;  
  bd,bb:array[1..38] of real;  
  bs:array[1..38] of string;  
  ry1,ry2:array[1..6,1..4] of extended;  
  cs1,cs:array[1..14] of string;  
  x,y1,y2:array[1..30] of real;  
  vy:array[1..4,1..38] of real;  
  ko:array[1..10] of real;  
  vm,pb,tb,mv,mp,xmi,xma,yml,yml2,yml,yml2,grx,gry1,gry2,kb,kc:real;  
  me0,me1,me2,me3,me4,va,i,j,n,tg,g1,g2,g3,mil,mi2:integer;  
  ex,g,b1,b2,b3:string;
```

{%I Tisk.pas}

Procedure CLS;

```
begin  
  TextBackground(Black);  
  Window(1,1,79,24);  
  ClrScr;  
end;
```

Procedure InKonst;

```
var ii:integer;  
  
begin  
  Window(1,1,79,23);  
  ClrScr;  
  TextColor(Blue);  
  GoToXY(25,2);writeln('ZADANI KONSTANT PRISTROJU');  
  GoToXY(20,3);writeln('-----');  
  writeln;  
  TextColor(Magenta);  
  write(' Zadej konstantu brzdy :');readln(kb);  
  writeln;  
  write(' Zadej konstantu clonky :');readln(kc);  
  writeln;  
  for ii:=1 to 8 do  
    begin  
      write(' Zadej konstantu tlakomeru',as[ii+17],':');  
      readln(ko[ii]);  
      writeln;  
    end;  
  write(' Zadej konstantu tlakomeru',as[27],':');  
  readln(ko[9]);  
  writeln;  
  write(' Zadej konstantu vzorce p(D2kin) dle potrubu :');  
  read(ko[10]);  
end;
```

Procedure Z0sy; {zadani promennych na osach}

label 2;

begin

2:g1:=0;g2:=0;g3:=0;

TextBackground(Blue);TextColor(Yellow);

Window(1,1,79,24);

ClrScr;

GoToXY(26,1);writeln('ZADANI VELICIN NA OSACH GRAFU');

TextColor(LightCyan);

For i:=1 to 38 do

begin

if i<=19 then

begin

GoToXY(5,i+2);writeln(i,' ',bs[i]);

end

else

begin

GoToXY(40,i-17);writeln(i,' ',bs[i]);

end;

end;

Window(2,21,78,24);

TextColor(Magenta);TextBackground(LightGray);

ClrScr;

GoToXY(10,1);writeln('Zadej typ grafu: 1. X-Y');

GoToXY(31,2);writeln('2. X-Y1,Y2');

GoToXY(10,4);write('Zadej cislo volby a stiskni <ENTER>:');readln(tg);

ClrScr;

GoToXY(15,1);write('Zadej velicinu na ose X (jeji cislo):');

readln(g1);

if tg=1 then

begin

GoToXY(15,2);write('Zadej velicinu na ose Y (jeji cislo):');

readln(g2);

g3:=0;

end

else

begin

GoToXY(15,2);write('Zadej velicinu na ose Y1 (jeji cislo):');

readln(g2);

GoToXY(15,3);write('Zadej velicinu na ose Y2 (jeji cislo):');

readln(g3);

end;

reset(so3);

for i:=1 to n do

begin

for j:=1 to 38 do

begin

readln(so3,bd[j]);

end;

if tg=1 then

begin

x[i]:=0;y1[i]:=0;

x[i]:=bd[g1];

y1[i]:=bd[g2];

end;

if tg=2 then

begin

x[i]:=0;y1[i]:=0;y2[i]:=0;

x[i]:=bd[g1];

y1[i]:=bd[g2];

y2[i]:=bd[g3];

end;

end;

close(so3);

for i:=1 to n do

begin

if x[i]<=0 then

```

CLS;
TextBackground(Red);TextColor(Black);
Window(10,8,60,16);
ClrScr;
GotoXY(5,15);writeln('      !!!! POZOR      !!!!');
GotoXY(7,15);writeln('  Nektera z hodnot zadannych na osu X');
GotoXY(8,15);writeln('    je rovna 0 - nelze provest regresi. ');
GotoXY(10,15);writeln('  Zadejte jinou velicinu na tuto osu. ');
Sound(1000);Delay(400);NoSound;
readln;
goto 2;
end;
if y1[i]<=0 then
begin
CLS;
TextBackground(Red);TextColor(Black);
Window(10,8,60,16);
ClrScr;
GotoXY(5,15);writeln('      !!!! POZOR      !!!!');
GotoXY(7,15);writeln('  Nektera z hodnot zadannych na osu Y1');
GotoXY(8,15);writeln('    je rovna 0 - nelze provest regresi. ');
GotoXY(10,15);writeln('  Zadejte jinou velicinu na tuto osu. ');
Sound(1000);Delay(400);NoSound;
readln;
goto 2;
end;
if tg=2 then
begin
if y2[i]<=0 then
begin
CLS;
TextBackground(Red);TextColor(Black);
Window(10,8,60,16);
ClrScr;
GotoXY(5,15);writeln('      !!!! POZOR      !!!!');
GotoXY(7,15);writeln('  Nektera z hodnot zadannych na osu Y2');
GotoXY(8,15);writeln('    je rovna 0 - nelze provest regresi. ');
GotoXY(10,15);writeln('  Zadejte jinou velicinu na tuto osu. ');
Sound(1000);Delay(400);NoSound;
readln;
goto 2;
end;
end;
end;
end;

```

Procedure LReg1; ($y=a \cdot b^x$)

```

var a1,a2,b1,b2,r1,r2,s1,s2,s3,sz,sx,z:extended;

begin
s1:=0;s2:=0;s3:=0;sx:=0;sz:=0;
for i:=1 to n do
begin
z:=ln(y1[i]);
s1:=s1+z*x[i];s2:=s2+x[i]*x[i];s3:=s3+z*z;
sx:=sx+x[i];sz:=sz+z;
end;
b1:=(n*s1-sx*sz)/(n*s2-sx*sx);
a1:=sz/n-(sx/n*b1);
r1:=(n*s1-sx*sz)/sqrt((n*s2-sx*sx)*(n*s3-sz*sz));
ry1[1,1]:=exp(a1);ry1[1,2]:=exp(b1);
ry1[1,4]:=r1;ry1[1,3]:=0;
if tg=2 then
begin
s1:=0;s2:=0;s3:=0;sx:=0;sz:=0;a2:=0;b2:=0;r2:=0;
for i:=1 to n do
begin
z:=ln(y2[i]);

```

```

sx:=sx+x[i];sz:=sz+z;
end;
b2:=(n*s1-sx*sz)/(n*s2-sx*sx);
a2:=sz/n-(sx/n*b2);
r2:=(n*s1-sx*sz)/sqrt((n*s2-sx*sx)*(n*s3-sz*sz));
ry2[1,1]:=exp(a2);ry2[1,2]:=exp(b2);
ry2[1,4]:=r2;ry2[1,3]:=0;
end;
end;

```

Procedure LReg2; { $y = a e^{(bx)}$ }

```

var a1,a2,b1,b2,r1,r2,s1,s2,s3,sz,sx,z:extended;

begin
s1:=0;s2:=0;s3:=0;sx:=0;sz:=0;
for i:=1 to n do
begin
z:=ln(y1[i]);
s1:=s1+z*x[i];s2:=s2+x[i]*x[i];s3:=s3+z*z;
sx:=sx+x[i];sz:=sz+z;
end;
b1:=(n*s1-sx*sz)/(n*s2-sx*sx);
a1:=sz/n-(sx/n*b1);
r1:=(n*s1-sx*sz)/sqrt((n*s2-sx*sx)*(n*s3-sz*sz));
ry1[2,1]:=exp(a1);ry1[2,2]:=b1;
ry1[2,4]:=r1;ry1[2,3]:=0;
if tg=2 then
begin
s1:=0;s2:=0;s3:=0;sx:=0;sz:=0;a2:=0;b2:=0;r2:=0;
for i:=1 to n do
begin
z:=ln(y2[i]);
s1:=s1+z*x[i];s2:=s2+x[i]*x[i];s3:=s3+z*z;
sx:=sx+x[i];sz:=sz+z;
end;
b2:=(n*s1-sx*sz)/(n*s2-sx*sx);
a2:=sz/n-(sx/n*b2);
r2:=(n*s1-sx*sz)/sqrt((n*s2-sx*sx)*(n*s3-sz*sz));
ry2[2,1]:=exp(a2);ry2[2,2]:=b2;
ry2[2,4]:=r2;ry2[2,3]:=0;
end;
end;

```

Procedure LReg3; { $y = a + b * x^{1/2}$ }

```

var a1,a2,b1,b2,r1,r2,s1,s2,s3,sz,sy,z:extended;

begin
s1:=0;s2:=0;s3:=0;sy:=0;sz:=0;
for i:=1 to n do
begin
z:=sqrt(x[i]);
s1:=s1+z*y1[i];s2:=s2+z*z;s3:=s3+y1[i]*y1[i];
sy:=sy+y1[i];sz:=sz+z;
end;
b1:=(n*s1-sy*sz)/(n*s2-sz*sz);
a1:=sy/n-(sz/n*b1);
r1:=(n*s1-sy*sz)/sqrt((n*s2-sz*sz)*(n*s3-sy*sy));
ry1[3,1]:=a1;ry1[3,2]:=b1;
ry1[3,4]:=r1;ry1[3,3]:=0;
if tg=2 then
begin
s1:=0;s2:=0;s3:=0;sy:=0;sz:=0;a2:=0;b2:=0;r2:=0;
for i:=1 to n do
begin
z:=sqrt(x[i]);

```

```

sy:=sy+y2[i];sz:=sz+z;
end;
b2:=(n*s1-sy*sz)/(n*s2-sz*sz);
a2:=sy/n-(sz/n*b2);
r2:=(n*s1-sy*sz)/sqrt((n*s2-sz*sz)*(n*s3-sy*sy));
ry2[3,1]:=a2;ry2[3,2]:=b2;
ry2[3,4]:=r2; ry2[3,3]:=0;
end;
end;

```

Procedure LReg4; { y=a+b/x }

```

var a1,a2,b1,b2,r1,r2,s1,s2,s3,sz,sy,z:extended;

begin
s1:=0;s2:=0;s3:=0;sy:=0;sz:=0;
for i:=1 to n do
begin
z:=1/x[i];
s1:=s1+z*y1[i];s2:=s2+z*z;s3:=s3+y1[i]*y1[i];
sy:=sy+y1[i];sz:=sz+z;
end;
b1:=(n*s1-sy*sz)/(n*s2-sz*sz);
a1:=sy/n-(sz/n*b1);
r1:=(n*s1-sy*sz)/sqrt((n*s2-sz*sz)*(n*s3-sy*sy));
ry1[4,1]:=a1;ry1[4,2]:=b1;
ry1[4,4]:=r1;ry1[4,3]:=0;
if tg=2 then
begin
s1:=0;s2:=0;s3:=0;sy:=0;sz:=0;a2:=0;b2:=0;r2:=0;
for i:=1 to n do
begin
z:=1/x[i];
s1:=s1+z*y2[i];s2:=s2+z*z;s3:=s3+y2[i]*y2[i];
sy:=sy+y2[i];sz:=sz+z;
end;
b2:=(n*s1-sy*sz)/(n*s2-sz*sz);
a2:=sy/n-(sz/n*b2);
r2:=(n*s1-sy*sz)/sqrt((n*s2-sz*sz)*(n*s3-sy*sy));
ry2[4,1]:=a2;ry2[4,2]:=b2;
ry2[4,4]:=r2;ry2[4,3]:=0;
end;
end;

```

Procedure LReg5; { y=a+bx }

```

var a1,a2,b1,b2,r1,r2,s1,s2,s3,sz,sx,z:extended;

begin
s1:=0;s2:=0;s3:=0;sx:=0;sz:=0;
for i:=1 to n do
begin
z:=y1[i];
s1:=s1+z*x[i];s2:=s2+x[i]*x[i];s3:=s3+z*z;
sx:=sx+x[i];sz:=sz+z;
end;
b1:=(n*s1-sx*sz)/(n*s2-sx*sx);
a1:=sz/n-(sx/n*b1);
r1:=(n*s1-sx*sz)/sqrt((n*s2-sx*sx)*(n*s3-sz*sz));
ry1[5,1]:=a1;ry1[5,2]:=b1;
ry1[5,4]:=r1;ry1[5,3]:=0;
if tg=2 then
begin
s1:=0;s2:=0;s3:=0;sx:=0;sz:=0;a2:=0;b2:=0;r2:=0;
for i:=1 to n do
begin
z:=y2[i];

```

```

sx:=sx+x[i];sz:=sz+z[i];
end;
b2:=(n*s1-sx*sz)/(n*s2-sx*sx);
a2:=sz/n-(sx/n*b2);
r2:=(n*s1-sx*sz)/sqrt((n*s2-sx*sx)*(n*s3-sz*sz));
ry2[5,1]:=a2;ry2[5,2]:=b2;
ry2[5,4]:=r2;ry2[5,3]:=0;
end;
end;

```

Procedure KReg;

```

var sx,sy1,sy2,s1,s2,s3,s4,s5,b1,b2,b3,yt,syx,syy,rx:extended;

```

```

begin
s1:=0;s2:=0;s3:=0;s4:=0;s5:=0;sx:=0;sy1:=0;syx:=0;syy:=0;rx:=0;
for i:=1 to n do
begin
s1:=s1+x[i]*y1[i];
s2:=s2+sqr(x[i]);
s3:=s3+exp(3*ln(x[i]));
s4:=s4+exp(4*ln(x[i]));
s5:=s5+x[i]*x[i]*y1[i];
sx:=sx+x[i];
sy1:=sy1+y1[i];
end;
b2:=(((n*s4-s2*s2)*(n*s1-sx*sy1))-((n*s3-s2*sx)*(n*s5-sx*sy1)))/((n*s2-sx*sx)*(n*s4-s2*s2))-sqr(n*s3-s2*sx);
b3:=(((n*s2-sx*sx)*(n*s5-s2*sy1))-((n*s3-s2*sx)*(n*s1-sx*sy1)))/((n*s2-sx*sx)*(n*s4-s2*s2))-sqr(n*s3-s3*s2);
b1:=sy1/n-b2*sx/n-b3*sqr(sx/n);
ry1[6,1]:=b1;ry1[6,2]:=b2;ry1[6,3]:=b3;
syx:=0;syy:=0;
for i:=1 to n do
begin
yt:=0;
yt:=b1+b2*x[i]+b3*x[i]*x[i];
syx:=syx+sqr(y1[i]-yt);
syy:=syy+sqr(y1[i]-sy1/n);
end;
syx:=syx/(n-1);
syy:=syy/(n-1);
rx:=sqrt(abs(1-syx/syy));
ry1[6,4]:=rx;
if tg=2 then
begin
s1:=0;s2:=0;s3:=0;s4:=0;s5:=0;sx:=0;sy2:=0;syx:=0;rx:=0;
for i:=1 to n do
begin
s1:=s1+x[i]*y2[i];
s2:=s2+sqr(x[i]);
s3:=s3+exp(3*ln(x[i]));
s4:=s4+exp(4*ln(x[i]));
s5:=s5+x[i]*x[i]*y2[i];
sx:=sx+x[i];
sy2:=sy2+y2[i];
end;
b2:=(((n*s4-s2*s2)*(n*s1-sx*sy2))-((n*s3-s2*sx)*(n*s5-sx*sy2)))/((n*s2-sx*sx)*(n*s4-s2*s2))-sqr(n*s3-s2*sx);
b3:=(((n*s2-sx*sx)*(n*s5-s2*sy2))-((n*s3-s2*sx)*(n*s1-sx*sy2)))/((n*s2-sx*sx)*(n*s4-s2*s2))-sqr(n*s3-s3*s2);
b1:=sy2/n-b2*sx/n-b3*sqr(sx/n);
ry2[6,1]:=b1;ry2[6,2]:=b2;ry2[6,3]:=b3;
syx:=0;syy:=0;
for i:=1 to n do
begin
yt:=0;
yt:=b1+b2*x[i]+b3*x[i]*x[i];
syx:=syx+sqr(y2[i]-yt);
syy:=syy+sqr(y2[i]-sy2/n);
end;
syx:=syx/(n-1);
syy:=syy/(n-1);

```

```

    ry2[6,4]:=rx;
  end;
end;

```

Procedure MinKor;

```

var mr1,mr2,b1,b2:extended;

begin
  mr1:=10;mr2:=10;mi1:=0;mi2:=0;
  for i:=1 to 6 do
    begin
      b1:=Abs(Abs(ry1[i,4])-1);
      if b1<mr1 then
        begin
          mi1:=i;
          mr1:=b1;
        end;
      if i=2 then
        begin
          b2:=Abs(Abs(ry2[i,4])-1);
          if b2<mr2 then
            begin
              mr2:=b2;
              mi2:=i;
            end;
          end;
        end;
      if abs(ry1[mi1,4])< 0.75 then mi1:=7;
      if abs(ry2[mi2,4])< 0.75 then mi2:=7;
    end;
  end;

```

Procedure TechData;

```

begin
  ClrScr;
  TextColor(Magenta);
  GotoXY(15,2);writeln('Zadani udaju o zkousce');
  writeln;
  TextColor(Brown);
  for i:=1 to 14 do
    begin
      write(cs[i], ' ');readln(cs[i]);
      writeln;
    end;
  vm:=0;pb:=0;tb:=0;va:=0;
  Val(cs[9],vm,i);
  Val(cs[11],pb,i);
  Val(cs[10],va,i);
  pb:=pb*10/75;
  Str(pb:7:3,cs[11]);
  Val(cs[12],tb,i);
end;

```

Procedure InDataKey;

```

var a:string;
    b:real;

begin
  TechData;
  ClrScr;
  writeln;
  writeln(' Zadej jmeno souboru,kde se ulozi data:');
  readln(a);
  writeln;

```

```

Assign(so1,a);
rewrite(so1);
for i:=1 to n do
begin
writeln;
for j:=1 to 27 do
begin
write(as[j]);read(b);
ad[j]:=b;
writeln(so1,ad[j]);
end;
end;
close(so1);
end;

```

Procedure InDataDisc;

```

var a,b:string;

begin
TechData;
ClrScr;
n:=0;
writeln;
writeln(' Zadej jmeno souboru,kde jsou ctena data:');
readln(a);Assign(so2,a);
writeln;
writeln(' Zadej jmeno souboru,kde se ulozi data:');
readln(b);Assign(so1,b);
rewrite(so1);
reset(so2);
while not eof (so2) do
begin
for i:=1 to 27 do
begin
readln(so2,ad[i]);
writeln(so1,ad[i]);
end;
n:=n+1;
end;
close(so1);close(so2);
end;

```

Procedure OprDat;

```

label 4;
var a:char;
r:real;

begin
ClrScr;
TextColor(Magenta);
GoToXY(20,3);writeln('OPRAVA DAT');
GoToXY(0,4);writeln('-----');
TextColor(Blue);
GoToXY(3,7);writeln(' Chcete-li radek opravit stisknete < A >');
GoToXY(3,9);writeln(' Nechcete-li ho opravit,');
GoToXY(3,10);writeln(' stisknete kteroukoliv jinou klavesu');
GoToXY(3,12);writeln(' Navrat do MENU stiskem < M >');
Delay(4500);
TextColor(Black);
writeln;writeln;
assign(so4,'c:\util\m\mopr.dat');
reset(so1);rewrite(so4);
while not eof (so1) do
begin
for i:=1 to 27 do
begin

```

```

readln(so1,ad[i]);
write(' ',as[i]);writeln(ad[i]:10:4);
Repeat
begin
a:=ReadKey;
if a='m' then goto 4;
if a='a' then
begin
ClrScr;
writeln('      Zadel novou hodnotu');
write(as[i]);
readln(ad[i]);
writeln(so4,ad[i]);
end
else
begin
writeln(so4,ad[i]);
end;
end;
until Keypressed=false;
end;
4: close(so1);close(so4);
reset(so4);rewrite(so1);
while not eof (so4) do
begin
for i:=1 to 27 do
begin
readln(so4,ad[i]);
writeln(so1,ad[i]);
end;
end;
close(so4);close(so1);
end;

```

Procedure Vypocet;

var ii:integer;

```

begin
Assign(so3,'c:\util\m\mvyp.dat');
rewrite(so3);
reset(so1);
for i:=1 to n do
begin
for j:=1 to 27 do
begin
readln(so1,ad[j]);
end;
for ii:=1 to 38 do
begin
bd[ii]:=0;
end;
bd[1]:=ad[1];bd[2]:=ad[2];bd[3]:=ad[3]*kb;
bd[4]:=ad[4];bd[5]:=ad[7];bd[6]:=ad[8];
bd[7]:=ad[9];bd[8]:=ad[10];bd[9]:=ad[11];
bd[10]:=ad[12];bd[11]:=ad[13];bd[12]:=ad[14];
bd[13]:=ad[15];bd[14]:=ad[16];bd[15]:=ad[17];
bd[16]:=ad[18]*ko[1];
bd[17]:=ad[19]*ko[2];
bd[18]:=ad[20]*ko[3];
bd[19]:=ad[21]*ko[4];
bd[20]:=ad[22]*ko[5];
bd[23]:=ad[23]*ko[6];
if bd[13]=0 then bd[33]:=0
else bd[33]:=(bd[23]+pb)*1000/(287.1*(bd[13]+273.15));
bd[25]:=ad[25]*ko[8];
if bd[25]=0 then mv:=0
else mv:=kc*(1-bd[25]*1000/(1800*pb))*sqrt(1000*bd[25]*bd[33]);

```

```

mp:=ad[6]/(ad[7]*1000);
if mv=0 then bd[21]:=0;
else bd[21]:=ko[10]*ad[12]/(bd[20]+pb)*mv*mv;
bd[22]:=bd[20]+bd[21]+pb;
bd[24]:=ad[24]*ko[7];
bd[26]:=ad[26];
bd[27]:=bd[3]*bd[1]*pi/30000;
bd[28]:=120*bd[27]/(vm*bd[1]);
bd[29]:=ad[5]*3600/(bd[27]*ad[4]);
bd[30]:=120*ad[5]/(va*bd[1]*ad[4]);
bd[31]:=120*ad[5]/(va*vm*bd[1]*ad[4]);
bd[32]:=pb*1000/(287.1*(tb+273.15));
bd[34]:=mv*120/(bd[1]*va);
bd[35]:=mv*120/(bd[1]*va*vm);
if mv=0 then bd[36]:=0;
else bd[36]:=mv/(14.32*mp);
bd[37]:=(bd[20]+pb)/(bd[19]+pb);
bd[38]:=ad[27]*ko[9];
for j:=1 to 38 do
begin
writeln(so3,bd[j]);
end;
end;
close(so1);close(so3);
end;

```

Procedure Vypis;

```

var gd,gm,g,ii,xg,yg:integer;
a,b:string;
ac:char;

```

```

begin
gd:=VGA;gm:=VGAhi;
InitGraph(gd,gm,'');
SetColor(Blue);SetBkColor(LightGray); { 1. strana protokolu }
ClearViewPort;
Rectangle(5,5,635,465);rectangle(5,50,635,85);rectangle(5,120,635,155);
rectangle(5,190,635,225);rectangle(5,260,635,295);line(5,335,635,335);
Line(5,48,635,48);Line(5,297,635,297);Line(5,333,635,333);
OutTextXY(40,315,'Cislo mereni');
Str(i,a);
if i>n then a:='';
OutTextXY(217,315,a);
Str(i+1,a);
if i+1>n then a:='';
OutTextXY(336,315,a);
Str(i+2,a);
if i+2>n then a:='';
OutTextXY(455,315,a);
Str(i+3,a);
if i+3>n then a:='';
OutTextXY(574,315,a);
For ii:=1 to 4 do
begin
g:=335+ii*25;
Line(5,g,635,g);
end;
For iis:=0 to 2 do
begin
g:=iis*157+160;
Line(g,50,g,295);
end;
for iis:=0 to 3 do
begin
g:=iis*119+159;
Line(g,295,g,465);
end;
SetTextStyle(0,0,1);SetTextJustify(1,0);

```

```

ii:=1;yg:=75;
while ii<=14 do
begin
  xg:=78;
  OutTextXY(xg,yg,cs[ii]);
  xg:=xg+157;
  OutTextXY(xg,yg,cs1[ii]);
  xg:=xg+157;ii:=ii+1;
  OutTextXY(xg,yg,cs[ii]);
  xg:=xg+157;
  OutTextXY(xg,yg,cs1[ii]);
  yg:=yg+35;
  ii:=ii+1;
end;
SetTextJustify(0,0);
for ii:=1 to 5 do
begin
  xg:=30;yg:=ii*25+328;
  OutTextXY(xg,yg,bs[ii]);
end;
SetTextJustify(1,0);
for ii:=1 to 5 do
begin
  xg:=217;yg:=ii*25+328;
  Str(vy[1,ii]:10:3,a);
  if i>n then a:=';
  OutTextXY(xg,yg,a);
  xg:=336;
  Str(vy[2,ii]:10:3,a);
  if i+1>n then a:=';
  OutTextXY(xg,yg,a);
  xg:=455;
  Str(vy[3,ii]:10:3,a);
  if i+2>n then a:=';
  OutTextXY(xg,yg,a);
  xg:=574;
  Str(vy[4,ii]:10:3,a);
  if i+3>n then a:=';
  OutTextXY(xg,yg,a);
end;
Repeat
begin
  ac:=ReadKey;
  if ac='1' then Tisk;
end;
until Keypressed=false;
ClearViewPort;
Rectangle(5,11,635,436);
For ii:=1 to 16 do
begin
  g:=11+ii*25;
  Line(5,g,635,g);
end;
For ii:=0 to 3 do
begin
  g:=ii*119+159;
  Line(g,11,g,436);
end;
for ii:=6 to 22 do
begin
  xg:=80;yg:=(ii-6)*25+28;
  OutTextXY(xg,yg,bs[ii]);
end;
SetTextJustify(1,0);
for ii:=6 to 22 do
begin
  xg:=217;yg:=(ii-6)*25+28;
  Str(vy[1,ii]:10:3,a);
  if i>n then a:=';
  OutTextXY(xg,yg,a);
  xg:=336;

```

{ 2. strana protokolu }

```

if i+1>n then a:=
OutTextXY(xg,yg,a);
xg:=455;
Str(vy[3,ii]:10:3,a);
if i+2>n then a:=
OutTextXY(xg,yg,a);
xg:=574;
Str(vy[4,ii]:10:3,a);
if i+3>n then a:=
OutTextXY(xg,yg,a);
end;
Repeat
begin
ac:=ReadKey;
if ac='1' then Tisk;
end;
until Keypressed=false;
ClearViewPort;
Rectangle(5,5,635,405);
For ii:=1 to 15 do
begin
g:=5+ii*25;
Line(5,g,635,g);
end;
For ii:=0 to 3 do
begin
g:=ii*119+159;
Line(g,5,g,405);
end;
for ii:=23 to 38 do
begin
xg:=80;yg:=(ii-23)*25+25;
OutTextXY(xg,yg,bs[ii]);
end;
SetTextJustify(1,0);
for ii:=23 to 38 do
begin
xg:=217;yg:=(ii-23)*25+25;
Str(vy[1,ii]:10:3,a);
if i>n then a:=
OutTextXY(xg,yg,a);
xg:=336;
Str(vy[2,ii]:10:3,a);
if i+1>n then a:=
OutTextXY(xg,yg,a);
xg:=455;
Str(vy[3,ii]:10:3,a);
if i+2>n then a:=
OutTextXY(xg,yg,a);
xg:=574;
Str(vy[4,ii]:10:3,a);
if i+3>n then a:=
OutTextXY(xg,yg,a);
end;
Repeat
begin
ac:=ReadKey;
if ac='1' then Tisk;
end;
until Keypressed=false;
CloseGraph;
end;

```

{3. strana protokolu }

Procedure Protokol;

var k:integer;

```

begin
ClrScr;
TextColor(Red);

```

```

GotoXV(0,4);writeln('Chcete-li pa...');
writeln('vytisknout tuto stranu protokolu');
writeln('stisknete klavesu < 1 >');
writeln;
writeln('nechcete-li vytisknout,');
writeln('stisknete kteroukoliv jinou klavesu.');
```

Delay(3500);

```

reset(so3);
i:=1;
while not eof (so3) do
begin
j:=1;
while j < 5 do
begin
for k:=1 to 38 do
begin
vy[i,k]:=0;
readln(so3,bd[k]);
vy[i,k]:=bd[k];
end;
j:=j+1;
end;
Vypis;
i:=i+4;
end;
close(so3);
end;
```

Procedure MinMax;

```

begin
xmi:=10000;xma:=0;ymi1:=10000;yma1:=0;ymi2:=10000;yma2:=0;
for i:=1 to n do
begin
if x[i]<xmi then xmi:=x[i];
if x[i]>xma then xma:=x[i];
if y1[i]<ymi1 then ymi1:=y1[i];
if y1[i]>yma1 then yma1:=y1[i];
if y2[i]<ymi2 then ymi2:=y2[i];
if y2[i]>yma2 then yma2:=y2[i];
end;
end;
```

Procedure OprOs;

```

var op:string;
o:real;
```

```

begin
ClrScr;
writeln('Chces-li zmenit krajni body na osach,');
writeln('po jejich vypisu stiskni <a> ');
writeln('jinak stiskni kteroukoliv jinou klavesu');
writeln('-----');
writeln;writeln;
TextColor(Magenta);
writeln(' X min:',xmi:10:3);
Repeat
begin
op:=ReadKey;
if op= 'a' then
begin
write(' Zadej novou hodnotu:');
readln(o);
xmi:=o;
end;
end;
```

```

end;
until Keypressed=false;
writeln(' X max:',xma:10:3);
Repeat
begin
op:=ReadKey;
if op= 'a' then
begin
write(' Zadej novou hodnotu:');
readln(o);
xma:=o;
end;
end;
until Keypressed=false;
writeln(' Y1 min:',ym1:10:3);
Repeat
begin
op:=ReadKey;
if op= 'a' then
begin
write(' Zadej novou hodnotu:');
readln(o);
ym1:=o;
end;
end;
until Keypressed=false;
writeln(' Y1 max:',yma1:10:3);
Repeat
begin
op:=ReadKey;
if op= 'a' then
begin
write(' Zadej novou hodnotu:');
readln(o);
yma1:=o;
end;
end;
until Keypressed=false;
if tg=2 then
begin
writeln(' Y2 min:',ymi2:10:3);
Repeat
begin
op:=ReadKey;
if op= 'a' then
begin
write(' Zadej novou hodnotu:');
readln(o);
ymi2:=o;
end;
end;
until Keypressed=false;
writeln(' Y2 max:',yma2:10:3);
Repeat
begin
op:=ReadKey;
if op= 'a' then
begin
write(' Zadej novou hodnotu:');
readln(o);
yma2:=o;
end;
end;
until Keypressed=false;
end;
end;

```

var gd,gm,xo,yo,ys,ac,gx,gy,gx1,gx2,gy1,gy2:integer;

p,py,z,gix:real;

ba,st1,st2,st3,st4:string;

ab:char;

begin

MinMax;

OprOsy;

Gd:=VGA;Gm:=VGAHi;

InitGraph(gd,gm,'');

if GraphResult <>GrOk then Halt (1);

SetBkColor(LightGray);SetColor(LightMagenta);

SetTextStyle(0,0,2);SetTextJustify(1,0);

OutTextXY(320,30,g);

SetColor(Blue);SetTextJustify(1,2);

SetTextStyle(0,0,1);

OutTextXY(320,425,b1); { osa x }

Line(100,390,540,390);

SetTextStyle(0,0,0);

gx:=(xma-xmi)/440;

xo:=100;p:=xmi;

while xo<540 do

begin

Line(xo,385,xo,395);

Str(p:9:3,ba);

OutTextXY(xo,403,ba);

xo:=xo+44;p:=p+(xma-xmi)/10;

Line(xo,385,xo,395);

Str(p:9:3,ba);

OutTextXY(xo,416,ba);

xo:=xo+44;p:=p+(xma-xmi)/10;

end;

Line(xo,385,xo,395);

Str(p:9:3,ba);

OutTextXY(xo,403,ba);

Line(100,60,100,390);

SetTextStyle(0,1,1);SetTextJustify(1,0);

OutTextXY(10,280,b2); { osa y1 }

gy1:=(yma1-ymi1)/330;

yo:=390;

SetTextStyle(0,0,0);SetTextJustify(2,0);

p:=ymi1;

while yo>=60 do

begin

Line(95,yo,105,yo);

Str(p:9:3,ba);

ys:=yo+5;

OutTextXY(90,ys,ba);

yo:=yo-33;p:=p+(yma1-ymi1)/10;

end;

if tg=2 then

begin

SetColor(Magenta); { osa y2 }

Line(540,60,540,390);

SetTextStyle(0,1,1);SetTextJustify(1,0);

OutTextXY(630,280,b3);

gy2:=(yma2-ymi2)/330;

yo:=390;

SetTextStyle(0,0,0);SetTextJustify(0,0);

p:=ymi2;

while yo>=60 do

begin

Line(535,yo,545,yo);

Str(p:9:3,ba);

ys:=yo+5;

OutTextXY(550,ys,ba);

yo:=yo-33;p:=p+(yma2-ymi2)/10;

end;

end;

SetColor(Blue);

Line(10,300,10,340);

Line(7,290,13,290);Line(10,287,10,293);

```

for i:=1 to n do
begin
  gx:=Round((x[i]-xmi)/grx)+100;
  gy:=390-Round((y1[i]-ymi1)/gry1);
  Line(gx-3,gy,gx+3,gy);Line(gx,gy-3,gx,gy+3);
end;
for i:=1 to n-1 do
begin
  gx1:=Round((x[i]-xmi)/grx)+100;
  gy1:=390-Round((y1[i]-ymi1)/gry1);
  gx2:=Round((x[i+1]-xmi)/grx)+100;
  gy2:=390-Round((y1[i+1]-ymi1)/gry1);
  Line(gx1,gy1,gx2,gy2);
end;
gix:=xmi;
while gix<=xma do
begin
  if mi1=1 then
  begin
    z:=exp(ln(ry1[1,1])+gix*ln(ry1[1,2]));
    gy:=390-Round((z-ymi1)/gry1);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,2)
    else gix:=gix+grx;
  end;
  if mi1=2 then
  begin
    z:=exp(ln(ry1[2,1])+gix*ry1[2,2]);
    gy:=390-Round((z-ymi1)/gry1);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,2)
    else gix:=gix+grx;
  end;
  if mi1=3 then
  begin
    z:=ry1[3,1]+ry1[3,2]*sqrt(gix);
    gy:=390-Round((z-ymi1)/gry1);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,2)
    else gix:=gix+grx;
  end;
  if mi1=4 then
  begin
    z:=ry1[4,1]+ry1[4,2]/gix;
    gy:=390-Round((z-ymi1)/gry1);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,2)
    else gix:=gix+grx;
  end;
  if mi1=5 then
  begin
    z:=ry1[5,1]+ry1[5,2]*gix;
    gy:=390-Round((z-ymi1)/gry1);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,2)
    else gix:=gix+grx;
  end;
  if mi1=6 then
  begin
    z:=ry1[6,1]+ry1[6,2]*gix+ry1[6,3]*gix*gix;
    gy:=390-Round((z-ymi1)/gry1);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,2)
    else gix:=gix+grx;
  end;
  gix:=gix+grx;
end;
if tg=2 then
begin
  SetColor(Magenta);
  SetLineStyle(1,0,1);
  Line(430,300,630,340);

```

```

for i:=1 to n do
begin
  gx:=Round((x[i]-xmi)/grx)+100;
  gy:=390-Round((y2[i]-ymi2)/gry2);
  PutPixel(gx,gy,5);Circle(gx,gy,3);
end;
for i:=1 to n-1 do
begin
  gx1:=Round((x[i]-xmi)/grx)+100;
  gy1:=390-Round((y2[i]-ymi2)/gry2);
  gx2:=Round((x[i+1]-xmi)/grx)+100;
  gy2:=390-Round((y2[i+1]-ymi2)/gry2);
  Line(gx1,gy1,gx2,gy2);
end;
SetColor(Red);
gix:=xmi;
while gix<=xma do
begin
  if mi2=1 then
  begin
    z:=exp(ln(ry2[1,1])+gix*ln(ry2[1,2]));
    gy:=390-Round((z-ymi2)/gry2);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,4)
    else gix:=gix+3*grx;
  end;
  if mi2=2 then
  begin
    z:=exp(ln(ry2[2,1])+gix*ry2[2,2]);
    gy:=390-Round((z-ymi2)/gry2);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,4)
    else gix:=gix+3*grx;
  end;
  if mi2=3 then
  begin
    z:=ry2[3,1]+ry2[3,2]*sqrt(gix);
    gy:=390-Round((z-ymi2)/gry2);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,4)
    else gix:=gix+3*grx;
  end;
  if mi2=4 then
  begin
    z:=ry2[4,1]+ry2[4,2]/gix;
    gy:=390-Round((z-ymi2)/gry2);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,4)
    else gix:=gix+3*grx;
  end;
  if mi2=5 then
  begin
    z:=ry2[5,1]+ry2[5,2]*gix;
    gy:=390-Round((z-ymi2)/gry2);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,4)
    else gix:=gix+3*grx;
  end;
  if mi2=6 then
  begin
    z:=ry2[6,1]+ry2[6,2]*gix+ry2[6,3]*gix*gix;
    gy:=390-Round((z-ymi2)/gry2);
    gx:=Round((gix-xmi)/grx)+100;
    if (gy>50) or (gy<390) then PutPixel(gx,gy,4)
    else gix:=gix+3*grx;
  end;
  gix:=gix+3*grx;
end;
if tg=1 then
begin

```

```

SetTextJustify(0,0);
SetColor(Green);
Line(40,465,80,465);
if mil=1 then
begin
Str(ry1[1,1]:8:3,st1);
Str(ry1[1,2]:8:3,st2);
Str(ry1[1,4]:8:3,st4);
OutTextXY(90,470,Concat(b2,'=',st1,'*',st2,'^',b1));
OutTextXY(550,470,Concat('r=',st4));
end;
if mil=2 then
begin
Str(ry1[2,1]:8:3,st1);
Str(ry1[2,2]:8:3,st2);
Str(ry1[2,4]:8:3,st4);
OutTextXY(90,470,Concat(b2,'=',st1,'*e^(',st2,'*',b1,')'));
OutTextXY(550,470,Concat('r=',st4));
end;
if mil=3 then
begin
Str(ry1[3,1]:8:3,st1);
Str(ry1[3,2]:8:3,st2);
Str(ry1[3,4]:8:3,st4);
OutTextXY(90,470,Concat(b2,'=',st1,'+',st2,'*',b1,'^(1/2)'));
OutTextXY(550,470,Concat('r=',st4));
end;
if mil=4 then
begin
Str(ry1[4,1]:8:3,st1);
Str(ry1[4,2]:8:3,st2);
Str(ry1[4,4]:8:3,st4);
OutTextXY(90,470,Concat(b2,'=',st1,'+',st2,'/',b1));
OutTextXY(550,470,Concat('r=',st4));
end;
if mil=5 then
begin
Str(ry1[5,1]:8:3,st1);
Str(ry1[5,2]:8:3,st2);
Str(ry1[5,4]:8:3,st4);
OutTextXY(90,470,Concat(b2,'=',st1,'+',st2,'*',b1));
OutTextXY(550,470,Concat('r=',st4));
end;
if mil=6 then
begin
Str(ry1[6,1]:8:3,st1);
Str(ry1[6,2]:8:3,st2);
Str(ry1[6,3]:8:3,st3);
Str(ry1[6,4]:8:3,st4);
OutTextXY(90,470,Concat(b2,'=',st1,'+',st2,'*',b1,'+',st3,'*',b1,'^2'));
OutTextXY(550,470,Concat('r=',st4));
end;
if mil=7 then
begin
OutTextXY(90,470,Concat('Pro',b2,' nebyla nalezena vhodna regresni krivka r<0.75'));
end;
end;
if tg=2 then
begin
SetTextJustify(0,0);
SetColor(Red);
SetLineStyle(1,1,2);
Line(40,460,80,460);
SetLineStyle(0,1,2);
SetColor(Green);
Line(40,450,80,450);
if mil=1 then
begin
Str(ry1[1,1]:8:3,st1);
Str(ry1[1,2]:8:3,st2);
Str(ry1[1,4]:8:3,st4);
OutTextXY(90,455,Concat(b2,'=',st1,'*',st2,'^',b1));

```

```

OutTextXY(550,455,Concat('r=',st4));
end;
if mi1=2 then
begin
Str(ry1[2,1]:8:3,st1);
Str(ry1[2,2]:8:3,st2);
Str(ry1[2,4]:8:3,st4);
OutTextXY(90,455,Concat(b2,'=',st1,' * e^(' ,st2,' * ',b1,' ')));
OutTextXY(550,455,Concat('r=',st4));
end;
if mi1=3 then
begin
Str(ry1[3,1]:8:3,st1);
Str(ry1[3,2]:8:3,st2);
Str(ry1[3,4]:8:3,st4);
OutTextXY(90,455,Concat(b2,'=',st1,' + ',st2,' * ',b1,' ^1/2'));
OutTextXY(550,455,Concat('r=',st4));
end;
if mi1=4 then
begin
Str(ry1[4,1]:8:3,st1);
Str(ry1[4,2]:8:3,st2);
Str(ry1[4,4]:8:3,st4);
OutTextXY(90,455,Concat(b2,'=',st1,' + ',st2,' / ',b1));
OutTextXY(550,455,Concat('r=',st4));
end;
if mi1=5 then
begin
Str(ry1[5,1]:8:3,st1);
Str(ry1[5,2]:8:3,st2);
Str(ry1[5,4]:8:3,st4);
OutTextXY(90,455,Concat(b2,'=',st1,' + ',st2,' * ',b1));
OutTextXY(550,455,Concat('r=',st4));
end;
if mi1=6 then
begin
Str(ry1[6,1]:8:3,st1);
Str(ry1[6,2]:8:3,st2);
Str(ry1[6,3]:8:3,st3);
Str(ry1[6,4]:8:3,st4);
OutTextXY(90,455,Concat(b2,'=',st1,' + ',st2,' * ',b1,' + ',st3,' * ',b1,' ^2'));
OutTextXY(550,455,Concat('r=',st4));
end;
if mi1=7 then
begin
OutTextXY(90,455,Concat('Pro',b2,' nebyla nalezena vhodna regresni krivka r<0.75'));
end;
SetColor(Red);
if mi2=1 then
begin
Str(ry2[1,1]:8:3,st1);
Str(ry2[1,2]:8:3,st2);
Str(ry2[1,4]:8:3,st4);
OutTextXY(90,465,Concat(b3,'=',st1,' * ',st2,' ^ ',b1));
OutTextXY(550,465,Concat('r=',st4));
end;
if mi2=2 then
begin
Str(ry2[2,1]:8:3,st1);
Str(ry2[2,2]:8:3,st2);
Str(ry2[2,4]:8:3,st4);
OutTextXY(90,465,Concat(b3,'=',st1,' * e^(' ,st2,' * ',b1,' ')));
OutTextXY(550,465,Concat('r=',st4));
end;
if mi2=3 then
begin
Str(ry2[3,1]:8:3,st1);
Str(ry2[3,2]:8:3,st2);
Str(ry2[3,4]:8:3,st4);
OutTextXY(90,465,Concat(b3,'=',st1,' + ',st2,' * ',b1,' ^1/2'));
OutTextXY(550,465,Concat('r=',st4));
end;

```

```

if mi2=4 then
begin
Str(ry2[4,1]:8:3,st1);
Str(ry2[4,2]:8:3,st2);
Str(ry2[4,4]:8:3,st4);
OutTextXY(90,465,Concat(b3,'=',st1,'+',st2,'/',b1));
OutTextXY(550,465,Concat('r=',st4));
end;
if mi2=5 then
begin
Str(ry2[5,1]:8:3,st1);
Str(ry2[5,2]:8:3,st2);
Str(ry2[5,4]:8:3,st4);
OutTextXY(90,465,Concat(b3,'=',st1,'+',st2,'*',b1));
OutTextXY(550,465,Concat('r=',st4));
end;
if mi2=6 then
begin
Str(ry2[6,1]:8:3,st1);
Str(ry2[6,2]:8:3,st2);
Str(ry2[6,3]:8:3,st3);
Str(ry2[6,4]:8:3,st4);
OutTextXY(90,465,Concat(b3,'=',st1,'+',st2,'*',b1,'+',st3,'*',b1,'^2'));
OutTextXY(550,465,Concat('r=',st4));
end;
if mi2=7 then
begin
OutTextXY(90,465,Concat('Pro',b3,' nebyla nalezena vhodna regresni krivka r(0.75)'));
end;
end;
Repeat
begin
ab:=ReadKey;
if ab='1' then Tisk;
end;
until Keypressed=false;
CloseGraph;
end;

```

```

begin {zacatek programu}

```

```

{zadavane hodnoty}

```

```

as[1]:=' n [1/min]';as[2]:=' nt [1/min]';as[3]:=' Mt [udaj]';
as[4]:=' Cas spotreby [s]';as[5]:=' Mnozstvi paliva M [g]';as[6]:=' Teplota oleje to=';
as[7]:=' Teplota vody tw=';as[8]:=' Teplota pred turbinou t(1T1)=';
as[9]:=' Teplota pred turbinou t(2T1)=';
as[10]:=' Teplota za turbinou t(T2)=';as[11]:=' Teplota pred dmychadlem t(D1)=';
as[12]:=' Teplota za dmychadlem t(D2)=';as[13]:=' Teplota pred mezich. t(*D3)=';
as[14]:=' Teplota za mezich. t(D3)=';as[15]:=' Teplota plnici t(p)=';
as[16]:=' Teplota vody pred chladicem t(w1)=';as[17]:=' Teplota vody za chladicem t(w2)=';
as[18]:=' Tlak oleje p(o) [udaj]';as[19]:=' dp(T1) [udaj]';
as[20]:=' dp(T2) [udaj]';as[21]:=' dp(D1) [udaj]';
as[22]:=' dp(D2) [udaj]';as[23]:=' dp(p) [udaj]';
as[24]:=' dp(ch) [udaj]';as[25]:=' Clonka dp [udaj]';
as[26]:=' Kourivost';as[27]:=' Filtr dp(F) [udaj]';

```

```

{vystupni hodnoty}

```

```

bs[1]:=' n [1/min]';bs[2]:=' n(t) [1/min]';bs[3]:=' Mt [Nm]';
bs[4]:=' t(o)';bs[5]:=' t(w)';bs[6]:=' t(1T1)';bs[7]:=' t(2T1)';
bs[8]:=' t(T2)';bs[9]:=' t(D1)';bs[10]:=' t(D2)';bs[11]:=' t(*D3)';
bs[12]:=' t(D3)';bs[13]:=' t(p)';bs[14]:=' t(w1)';bs[15]:=' t(w2)';
bs[16]:=' p(o) [MPa]';bs[17]:=' dp(T1) [kPa]';bs[18]:=' dp(T2) [kPa]';
bs[19]:=' dp(D1) [kPa]';bs[20]:=' dp(D2) [kPa]';
bs[21]:=' p(D2kin.) [kPa]';bs[22]:=' p(D2celk.) [kPa]';
bs[23]:=' dp(p) [kPa]';bs[24]:=' dp(ch) [kPa]';
bs[25]:=' dp [kPa]';bs[26]:=' k [%]';bs[27]:=' Pe [kW]';
bs[28]:=' p(e) [MPa]';bs[29]:=' m(pe) [g/kWh]';
bs[30]:=' m(px) [g/cy]';bs[31]:=' m(px) [g/cy.l]';
bs[32]:=' Ro(v0) [kg/m^3]';bs[33]:=' Ro(vpl) [kg/m^3]';
bs[34]:=' m(vx) [kg/cy]';bs[35]:=' m(vx) [kg/cy.l]';

```

bs[36]:= 'Lambda';bs[37]:= 'Stlacení Pi';bs[38]:= 'dp(f) [kPa]';

{udaje o motoru a zkousce}

```
cs[1]:= 'Typ motoru:';  
cs[2]:= 'Jmenovity výkon:';  
cs[3]:= 'Jmenovite otacky:';  
cs[4]:= 'Vstrik. cernadlo:';  
cs[5]:= 'Trysky:';  
cs[6]:= 'Turbodmychadlo:';  
cs[7]:= 'Vyr.cislo motoru:';  
cs[8]:= 'Predvstrik:';  
cs[9]:= 'Zdvihovy objem:';  
cs[10]:= 'Pocet valcu:';  
cs[11]:= 'Barometricky tlak:';  
cs[12]:= 'Teplota vzduchu:';  
cs[13]:= 'Vlhkost vzduchu:';  
cs[14]:= 'Datum:';
```

{ konstanty pristroju }

```
ko[1]:=0.0981;  
ko[2]:=9.81;  
ko[3]:=0.00981;  
ko[4]:=0.00981;  
ko[5]:=0.1333;  
ko[6]:=0.1333;  
ko[7]:=0.00981;  
ko[8]:=0.00981;  
ko[9]:=0.00981;  
ko[10]:=9.7;  
kb:=6.66667;  
kc:=3.51026e-3;
```

```
CLS;  
TextBackground(LightGray);  
Window(12,6,72,20);  
ClrScr;  
TextBackground(Blue);TextColor(Yellow);  
window(10,5,70,19);  
ClrScr;  
GoToXY(25,7);writeln('DOBRY DEN !');  
Delay(4000);  
ClrScr;  
GoToXY(19,2);writeln('NEKOLIK UPOZORNENI :');  
GoToXY(15,3);writeln('-----');  
TextColor(White);  
GoToXY(2,5);writeln('- navrat do menu vzdy pomoci <ENTER>');  
GoToXY(2,7);writeln('- na disku musi byt vytvoreny adresare C:\UTIL\M\MVVF.DAT');  
GoToXY(2,9);writeln('      a C:\UTIL\M\MOPR.DAT');  
GoToXY(2,11);writeln('- pri vypoctu doporucuji ukladat data do C:\UTIL\M\MOTOR.DAT');  
readln;
```

```
CLS;  
TextBackground(LightGreen);TextColor(Blue);  
InKonst;
```

```
1: CLS;  
TextBackground(LightGray);  
Window(17,6,67,23);  
ClrScr;  
TextBackground(LightGreen);  
Window(15,5,65,22);  
ClrScr;  
TextColor(Magenta);  
GoToXY(20,3);writeln('HLAVNI MENU');  
TextColor(Blue);  
GoToXY(8,7);writeln('1. Zadani dat');  
GoToXY(8,9);writeln('2. Oprava dat');  
GoToXY(8,11);writeln('3. Kresleni grafu');  
GoToXY(8,13);writeln('4. Vypis protokolu');  
GoToXY(8,15);writeln('5. Ukonceni programu');  
GoToXY(8,18);write('Zadej cislo volby a stiskni <ENTER>');  
readln(mo0);
```

```

if me0=1 then
  begin
    TextBackground(LightGray);TextColor(Magenta);
    Window(10,7,60,17);
    ClrScr;
    GoToXY(20,2);writeln('ZADANI DAT');
    TextColor(Black);
    GoToXY(4,5);writeln('1. Cteni z disku nebo diskety');
    GoToXY(4,7);writeln('2. Zadani z klavesnice');
    GoToXY(7,10);write('Zadej cislo volby a stiskni <ENTER>:');
    readln(me1);
    if me1=2 then InDataKey;
    if me1=1 then InDataDisc;
    goto 1;
  end;
if me0=2 then OprDat;
if me0=3 then
  begin
3:   TextBackground(Cyan);TextColor(Magenta);
      Window(17,8,75,24);
      ClrScr;
      GoToXY(20,2);writeln('KRESLENI GRAFU');
      TextColor(Blue);
      GoToXY(5,6);writeln('1. Vyber velicin na osach');
      GoToXY(5,8);writeln('2. Popis os a grafu');
      GoToXY(5,10);writeln('3. Provedeni regrese');
      GoToXY(5,12);writeln('4. Kresleni a tisk grafu');
      GoToXY(10,16);write('Zadej cislo volby a stiskni <ENTER>:');
      readln(me3);
      TextColor(Red);
      if me3=1 then
        begin
          Vypocet;
          ZOs;
          goto 3;
        end;
      if me3=2 then
        begin
          ClrScr;
          writeln('Zadej nazev grafu :');
          readln(g);
          writeln(' Popis osy x je:',bs[g1]);
          writeln(' Popis osy y1 je:',bs[g2]);
          if tg=2 then
            begin
              writeln(' Popis osy y2 je:',bs[g3]);
            end;
          b1:=bs[g1];b2:=bs[g2];b3:=bs[g3];
          readln;
          goto 3;
        end;
      if me3=3 then
        begin
          for i:=1 to 4 do for j:=1 to 6 do
            begin
              ry1[j,i]:=0;
              ry2[j,i]:=0;
            end;
          LReg1;LReg2;LReg3;LReg4;LReg5;KReg;
          MinKor;
          goto 3;
        end;
      if me3=4 then
        begin
          ClrScr;
          GoToXY(15,8);writeln('Chcete-li graf vytisknout');
          GoToXY(15,10);writeln('      stisknete <1>');
          Delay(3000);
          Paint;
        end;
      goto 1;
    end;

```

```

if me0=4 then
begin
Vypocet;
Protokol;
end;
if me0=5 then
begin
TextBackground(LightGray);
Window(22,11,52,14);
ClrScr;
TextBackground(Red);TextColor(Yellow);
window(20,10,50,13);
ClrScr;
GotoXY(3,2);writeln('Chcete opravdu skoncit A/N');
Repeat
begin
ex:=ReadKey;
if ex='a' then EXIT;
if ex='n' then goto 1
end;
until Keypressed=false;
end;
goto 1;
end. {konec programu}.

```